Frühsignale für Änderungen von Konjunkturindikatoren durch Analysen von Big Data

Early Signals for changes of Economic Indicators using Big Data Analysis

Bachelorarbeit

Zur Erlangung des akademischen Grades

Bachelor of Science (B.Sc.)

in Volkswirtschaftslehre

Humboldt-Universität zu Berlin

Wirtschaftswissenschaftliche Fakultät

Ladislaus von Bortkiewicz Chair of Statistics

vorgelegt von

Daniel Jacob

Matrikel-Nr. 553770

Prüfer: Prof. Dr. W. Härdle

Betreuer: Lukas Borke

Berlin, 21. Oktober 2015

Abstract

In Zeiten, in denen immer mehr Daten über einzelne Individuen gesammelt werden, liegt es nahe, dass diese für die ökonomische Forschung nicht mehr unberücksichtigt bleiben. Die Anzahl an Publikationen welche sich mit dem Thema *Data Mining* und *Big Data* beschäftigen steigt von Jahr zu Jahr rapide an. Diese Arbeit gibt den aktuellen Forschungsstand in diesem Gebiet wieder und zeigt, welche Schritte notwendig sind, um mit solchen Datensätzen zu arbeiten. Dafür werden Methoden zur Strukturierung der Daten aufgezeigt und bewertet. Weiter werden statistische Verfahren erläutert und angewendet, welche die Einbindung der Datensätze erlaubt. Ziel ist es zu zeigen, dass bereits einfache Zeitreihenmodelle ausreichen, um die Nützlichkeit solcher nicht amtlichen Datengrundlagen auf Konjunkturindikatoren zu beschreiben. Es wird jedoch auch gezeigt, dass bei einer fast unbegrenzten Anzahl an möglichen Regressoren teils nicht kausale Zusammenhänge sehr leicht darzustellen sind. Daher stellt der Umgang, speziell die Selektion solcher Datenmengen wohl den schwierigsten Teil in diesem Forschungsbereich dar.

Schlagwörter: *Data Mining, Big Data, Forschungsstand, Dimensionsreduktion, LSA, Clustering, Random Forest, Bayesian Structural Time Series*

Inhaltsverzeichnis

Abkürzungsverzeichnis

BSTS	Bayesian Structural Time Series
CART	Classification and Regression Trees
ESI	Economic Sentiment Indicator
EVD	Eigenvalue Decomposition
FRM	Financial Risk Meter
GFK	Gesellschaft für Konsumforschung
LASSO	Least Absolute Shrinkage and Selection Operator
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
OLS	Ordinary Least Squares (Kleinste Quadrate)
PCA	Principal Component Analysis
SFB 649	Sonderforschungsbereich 649 der HU-Berlin
SVD	Singular Value Decomposition
TDM	Term-Dokument-Matrix

Abbildungsverzeichnis

Tabellenverzeichnis

Formelverzeichnis

1 Einleitung

Prognosen über die Entwicklung einer Ökonomie zu erstellen ist eine der Hauptaufgaben sowohl in Ministerien, Wirtschaftsinstituten als auch in der universitären Forschung. Zentrale Entscheidungen der Politik orientieren sich an den wahrscheinlichen Entwicklungen einer Volkswirtschaft. Doch durch welche Faktoren ändert sich die Ökonomie? Es sind die Menschen, welche jeden Tag hunderte von Entscheidungen treffen. Vom Kauf einer Cola für einen Euro, über das neue Auto für 40.000 Euro, bis hin zum Eigenheim für 400.000 Euro. Diese überwiegend monetären Entscheidungen werden indirekt von der amtlichen Statistik erfasst und daraus Daten erzeugt, welche konjunkturelle Entwicklungen in eine gewisse Richtung aufzeigen. Zwei Kriterien müssen dafür allerdings erfüllt sein. Zum einen muss der Kauf bereits abgeschlossen sein und zum anderen in der Regel das Quartal oder zumindest der Monat vorbei sein. Selbst dann kann es noch Wochen dauern, bis offizielle Änderungsraten veröffentlicht werden. Da jedoch gerade die Nachfrage ein wichtiges Warnsignal für ökonomische Schwankungen darstellen kann, ist dies oft zu spät. Wie kann man nun also früher an relevante Daten kommen? Laut einer Untersuchung der GFK in Kooperation mit Google (2011) lässt sich ein positiver Zusammenhang zwischen dem Preis eines Produkts und dem für den Kauf erbrachten Zeitaufwand zur Produktrecherche herstellen.

Weiter zeigt eine Studie von Würdinger (2015) für das TNS Infratest, dass die meisten Befragten zur Recherche das Internet und eine Suchmaschine, benutzen. Der Betreiber Google steht mit einem Marktanteil von 94% für Deutschland und 64% für die USA in diesem Segment an oberster Stelle (Statista-Das-Suchmaschinenportal, 2015). Da sowohl das Suchverhalten als auch dessen Frequenz über Google ausgelesen werden können, wäre dies ein direkter Weg, um mit Suchbegriffen frühe Signale für ein verändertes Verhalten seitens der Verbraucher zu überprüfen.

Wörter und Verhaltensweisen gehen dabei in einander ein. Zum einen kann der Einfluss von Wörtern in Medien wie Zeitungen oder auch Fernsehen dazu führen, dass Menschen ihre Handlungen verstärken oder gar stoppen. Verhaltensweisen können jedoch auch durch andere äußere Impulse beeinflusst werden und sich dann in Suchanfragen widerspiegeln. Wenn in Zeitungen häufig von Inflation die Rede ist, könnte dies dazu führen, dass materielle Anschaffungen vorgezogen werden. Bei Deflation erwartet die gängige Theorie genau den gegensätzlichen Effekt. Wird zum Beispiel verstärkt nach „Autoversicherung“ gesucht,

könnte dies ein Anzeichen dafür sein, dass demnächst mehr Fahrzeuge verkauft werden und somit das Bruttoinlandsprodukt steigen. Doch nicht nur das Kaufverhalten kann dadurch prognostiziert werden. Ein zweiter Anwendungsbereich ist nach Häufigkeiten von ökonomisch relevanten Wörtern bei Google zu suchen. Diese können dann relevant sein, wenn sich durch Suchverhalten sowohl in der Frequenz als auch geographisch Verhaltensweisen unterstellen lassen, welche für eine Volkswirtschaft relevant sind. Wenn diese eben ein aktuelles Interesse oder gar eine Angst einer Gruppe aufzeigen. Der Begriff „Arbeitsamt“ könnte bei überproportionaler Benutzung ein erstes Indiz dafür sein, dass sich verstärkt Menschen Sorgen um ihre Anstellung machen und deshalb Informationen sammeln.

In der folgenden Arbeit werde ich Verfahren zur Datenermittlung, der Bearbeitung und Visualisierung vorstellen. Darauffolgend werden verschiedene statistische Methoden zum Auswerten von Daten aufgezeigt und bewertet. Im direkten Übergang wende ich einige Methoden mit strukturierten Daten an, um deren Güte und Signalqualität auf ausgewählte Konjunkturindikatoren zu berechnen.

2 Aktueller Forschungsstand

Der Begriff Big Data wird in vielen Publikationen durch drei „V's“ beschrieben (Dumbill, 2012; Press, 2013). Dabei steht „Volume“ für die große Menge der Daten, sowohl an Beobachtungen im Zeitverlauf als auch an Merkmalen (z.B. Wörter). Weiter besitzen diese Daten eine hohe Umlaufgeschwindigkeit („Velocity“), welche sich besonders im Internet und dort in Sozialen Netzwerken bemerkbar macht (Sagiroglu and Sinanc, 2013). „Variety“ (Vielfalt) beschreibt einen eindeutigen Nachteil bei solch großen Datenmengen - sie sind in der Regel unstrukturiert. Dies kann zum einen daran liegen, dass sie aus verschiedenen Quellen stammen oder aber vorerst keine eindeutige Trennung zwischen relevanten und irrelevanten Daten vorgenommen werden kann. Letzteres trifft vor allem auf Texten aus Zeitungen zu, sodass hier als erster Schritt für die jeweilige Forschungsfrage relevante Informationen herausgezogen werden müssen (Einav and Levin, 2013). Seit 2012 gibt es noch ein viertes V: „Veracity“(Wahrhaftigkeit) (IBM, 2014). Auf diesem Aspekt liegt das Hauptaugenmerk dieser Arbeit. Wie verlässlich sind Daten welche sich fern ab jeglicher amtlichen Statistik befinden? Wie genau lassen sich damit Vorhersagen machen? Es geht hierbei also um das Erkennen von Zusammenhängen, von Mustern und Bedeutungen.

Schwierigkeiten im Umgang mit Big Data bestehen vor allem darin, dass die gängigen Prognoseprogramme weder mit der Größe, der Geschwindigkeit noch der Komplexität der Daten umgehen können (Madden, 2012). Um dieses Problem zu beheben, müssen die Daten zuerst strukturiert werden (Arribas-Bel, 2014). Es müssen also sinnvolle Prädiktoren ausgewählt werden. Dies kann manuell durch eine visuelle Darstellung aller und der dann gezielten Löschung nicht echter Variablen vorgenommen werden (Varian, 2014). Eine andere Möglichkeit ist, vor allem in großen Textdateien, eine automatische Dimensionsreduktion vorzunehmen. Hierbei werden Corpota erstellt, welche Wörter in vorher definierte Kategorien zusammenfassen (Mao et al., 2010; Bartenhagen, 2013). Sind die Variablen auf ca. 100-200 begrenzt, wie z.B. der Output von Google Correlate Tabellen, können diese direkt mit verschiedenen Verfahren, wie etwa dem *Bayesian Information Criterion (BIC)* oder auch *Akaike Information Criterion (AIC)* auf ihre Güte getestet werden. Castle et al. (2009) beschreiben und vergleichen dafür 21 verschiedene Methoden. Auf Google Suchbegriffe wenden Choi and Varian (2012) mit einem saisonalen AR-Modell eine der Methoden an. Zur Prognose des BIPs von Deutschland mit einem großen Panel an vierteljährlichen Zeitreihen übertrifft laut Schumacher (2007) die Verwendung eines Faktor-Modells[1] mit statischen und dynamischen Hauptkomponenten und Teilraum-Algorithmus die Prognosequalität der AR-Methode. In einem weiteren Paper wurde erstere Methode ausgedehnt (Schumacher and Breitung, 2008). Um das Problem des „overfittings“ (N>T) zu vermeiden, wird immer mehr auf Baum-Modelle (*Tree-Models*) gesetzt. Diese überzeugen durch ihre gute Prognosequalität wenn die Datensätze sehr groß sind (Perlich et al., 2003). Eine Weiterentwicklung davon ist das Bilden von mehreren Bäumen innerhalb des Modells.[2] Diese liefern ebenso bessere Ergebnisse als einfache AR Modelle (Biau and D'Elia, 2009). Besonders bei nicht linearen Daten können sehr gute Out-of-Sample Ergebnisse produziert werden (Howard and Bowles, 2012).

Um Variablen in linearen Modellen zu selektieren eignet sich z.B. das Least Absolute Shrinkage and Selection Operator (LASSO) Verfahren. Hierbei werden im Idealfall unbrauchbare Variablen genau gleich Null geschaltet, was die Auswahl guter Prädiktoren erleichtert (Doornik and Hendry, 2015). Eine weitere Methode, um Variablen sinnvoll auszuwählen, ist die Spike-and-Slap-Regression[3], bei der einzelne Variablen in mehreren Durchläufen entweder gleich Null oder Eins geschaltet werden und diesen dann durch eine

[1] Package für R: FactoMineR
[2] Package für R: RandomForest
[3] Package für R: spikeslab

Verknüpfung des vorherigen Durchgangs eine Wahrscheinlichkeit für ihre Teilnahme (= 1) zugeordnet wird (Ishwaran et al., 2013). Oben genannte Verfahren zur Variablenauswahl beziehen sich auf Querschnittsdaten, das heißt, diese unterliegen in der Regel einer unabhängigen Verteilung. Bei Zeitreihenanalysen empfiehlt Varian (2014) die Verwendung der Bayesian Structural Time Series (BSTS) Methode.[1] Diese ist konzipiert, um bei einer großen Anzahl an Prädiktoren (e.g. N > 1000) sowohl das Problem des „overfittings“ als auch fälschliche Korrelationen zu beheben.

Oben genannte Methoden finden in Verbindung mit großen Datenmengen in der Forschung immer häufiger Anwendung. De Mol et al. (2008) verwenden verschiedene makroökonomische Daten, um Vorhersagen über die industrielle Produktion und dem Konsumentenpreisindex (CPI) zu erstellen. Die gleichen zwei Indikatoren werden vier Jahre später durch Giovannelli (2012) mit 259 Prädiktoren und einer Principal Component Analyse (PCA) dargestellt. Altissimo et al. (2010) benutzen monatlich akkumulierte große Datenmengen, um Prognosen über das BIP der Eurozone zu berechnen. Banerjee et al. (2014) erweitern den Datensatz auf 90 verschiedene monatliche Zeitreihen, um Vorhersagen über das BIP in Deutschland zu geben. Ouysse (2013) nutzt große Paneldaten, um die Inflation in den Vereinigten Staaten zu schätzen. Iselin and Siliverstovs (2013) greifen den vom Economist (1998) entwickelten R-Word Index erneut auf und versuchen durch Auswerten der Häufigkeit des Wortes „Rezession“ in der Zeitung Handelsblatt eine Korrelation zum BIP Wachstum für Deutschland zu finden. Erweitert wird dies durch Ammann et al. (2014). Sie analysieren Zeitungen nach spezifischen Wörtern, um Kursschwankungen an der Börse zu prognostizieren.

Auch benutzen immer mehr Forscher Daten von Google, um verschiedene ökonomische Indikatoren abzubilden. Askitas and Zimmermann (2009) werten Suchhäufigkeiten nach „Arbeitsamt“ und „Job Börse“ aus, um Änderungen in der Arbeitslosigkeit in Deutschland zu deuten. Preis et al. (2013) stellen die These auf, dass das Verhalten von Menschen durch Suchanfragen gedeutet werden kann und somit vorhersagbare Auswirkungen auf Handelsstrategien an der Börse hat. Scott and Varian (2014b) benutzen sowohl Google Trends als auch Google Correlate, um mit einer BSTS Methode Rückschlüsse auf Arbeitslosengelder und Einzelhandelsumsätze zu führen. In einem Working Paper mit dem Titel „A Hands-on Guide to Google Data“ geben Stephens-Devidowitz and Varian (2015) einen Überblick über die Verwendungsmöglichkeiten der Daten von Google. Dabei zei-

[1] Package für R: bsts

gen sie unter anderem eine Korrelation zwischen Suchbegriffen und Häuserpreisen in den Vereinigten Staaten.

3 Datengrundlagen – Deskriptive Statistik

Um verschiedene Vorgehensweisen zum Sammeln und Bearbeiten von großen Datensätzen aufzuzeigen stammen die Datensätze aus unterschiedlichen Quellen und besitzen verschiedene Formate. Nachrichtentexte kommen von der NASDAQ Community Platform für den Zeitraum Oktober 2009 bis Oktober 2014. Die webbasierten Daten stammen von Google Trends, beginnend im Januar 2004 bis Juni 2015. Bei Google Trends Daten wird zwischen Suchbegriffen für Deutschland allgemein, dessen Bundesländer und für die USA als Ganzes unterschieden. Als Konjunkturvariablen wird das Jahreswachstum des Bruttoinlandsprodukts nach Berechnungen des Statistischen Bundesamts verwendet. Diese Daten liegen in quartalsweiser Unterteilung von Q1/2001 bis Q1/2015 vor. Ferner werden die Arbeitslosenquote je Monat im Zeitraum 01/2005 bis 05/2015 ebenfalls vom Statistischen Bundesamt und ein vom Sonderforschungsbereich der Humboldt-Universität entwickelter Risk-Meter von 07/2007 bis 07/2015 verwendet.

3.1 Konjunkturdaten der amtlichen Statistik

Die quartalsweisen Jahreswachstumsraten des BIP für Deutschland werden in Abbildung A.1 gezeigt. Aufgrund der Euroeinführung im Jahr 2000 beginnt die Zeitreihe erst ab dem Jahr 2001, um eventuelle Änderungsraten welche auf diese Umstellung zurückzuführen sind zu umgehen. Hervorzuheben sind zwei große Wendepunkte, der erste beginnend im Jahr 2008 mit einem maximalen Negativwachstum von -6,8%, welches die große Rezession zeigt. Ein zweiter nennenswerter Abschnitt beginnt Anfang 2011 mit einem Abschwung von +6% auf -1% während der Euro-Krise. Ein Boxplot zeigt in Abbildung A.4 die deskriptive Statistik.

Die Zeitreihe der Arbeitslosenquote für Deutschland beginnt im Januar 2005 und endet im Dezember 2014. Da die Daten monatlich zur Verfügung stehen, ergeben sich 120 Beobachtungen (Siehe Boxplot A.5). Ein zweiter Datensatz beschreibt die Arbeitslosenquote für die einzelnen Bundesländer im jährlichen Zeitraum von 2005 bis 03.2015. Hier wurden für das Jahr 2015 extra nur die Wintermonate verwendet, um saisonale Schwankungen aufzuzeigen. Abbildung A.2 zeigt den zeitlichen Verlauf.

3.2 NASDAQ News

Die *Financial News Articles* stammen von der Nasdaq Community Plattform und werden vom *Research Data Center* (RDC) der Humboldt-Universität zu Berlin bereitgestellt. Das verwendete Rohformat dieser Daten ist eine dokumentenbasierte-Ordnerstruktur und liegt als .txt Datei vor. Jeder Artikel befindet sich in einer extra Datei, welche zur späterer Verarbeitung zusammengefasst werden. Insgesamt gibt es 116.691 Artikel in einem Zeitraum von 10.2009 bis 10.2014.

3.3 Financial Risk Meter

Der *Financial Risk Meter* (FRM) ist ein vom Sonderforschungsbereich (SFB649) der Humboldt-Universität zu Berlin entwickelter Index (*SFB-Risk-Index*) zum Messen von systemischen Krisen in den USA. Dieser liegt in täglicher Basis vom Zeitraum 07.2007 bis 07.2015 vor. Zur Anwendung auf verschiedene Methoden wurde die tägliche Basis auf eine monatliche durch Bildung des arithmetischen Mittels hochgerechnet. Abbildung A.3 zeigt den zeitlichen Verlauf.

3.4 Google Trends

Google Trends zeigt die Häufigkeitsverteilung eines Suchbegriffes innerhalb eines Zeitraums an. Dazu wird ein prozentualer Anteil aller Suchanfragen analysiert. Die Daten werden normalisiert, i.e. es wird jede Häufigkeit eines bestimmten Begriffes durch die Gesamtzahl der Suchanfragen geteilt. Dies geschieht ebenso auf geographischer Ebene, um zu verhindern, dass Orte welche ein hohes Suchvolumen haben immer an erster Stelle stehen. Das höchste relative Suchaufkommen bekommt dann den Wert 100 und alle in einem Zeitraum verbleibenden Daten richten sich an diesem Index aus. Zu beachten ist hierbei, dass wenn ein Suchbegriff während eines Zeitraums in seiner Frequenz abfällt, es eben nicht heißen muss, dass weniger nach diesem gesucht haben, sondern evtl. die gesamten Suchanfragen nach allen anderen Begriffen zugenommen haben (Google, 2015).

Bei der Abfrage kann nach einem einzelnen Begriff, zusammenhängenden Wörtern und auch nach Kategorien gesucht werden. Weiter können verschiedene Filter eingestellt werden, wie z.B. die relative Anzahl eines Begriffes in einer bestimmten Kategorie, innerhalb eines Landes, eines Bundeslandes bis hin zu einer einzelnen Stadt. Für die USA stehen als weitere Unterteilung auch noch Landkreise (metro areas) zur Verfügung. Der

zeitliche Verlauf der Suchanfragen beginnt frühestens Anfang 2004 und ist bei einer Eingrenzung von bis zu drei Monaten in täglicher, von bis zu drei Jahren in wöchentlicher und danach in monatlicher Periodizität unterteilt. Mit dem Package *okugami79* für *R* ist es jedoch möglich, auch längere Zeitperioden in wöchentlichen Daten auszugeben (okugami79, 2013). Seit Juli 2015 ist es nun auch möglich die Daten in Echtzeit auszuwerten. Die kleinstmögliche Unterteilung hierbei ist eine Stunde.

Weiter hat Google Trends einen Schwellwert. Liegt die totale Anzahl nach einem Suchbegriff unter diesem wird der Wert 0 ausgegeben. Dieser kann evtl. durch eine Spezifizierung in Zeit und oder Ort umgangen werden. Wie bereits erwähnt wird zur Analyse nur eine Stichprobe aller Suchanfragen verwendet. Dies kann dazu führen, dass die Ergebnisse variieren. Um dies auszugleichen ist es sinnvoll, einen Mittelwert aus Suchanfragen aus verschiedenen Tagen (führt zu verschiedenen Stichproben) zu bilden. Ferner gilt zu beachten, dass Suchbegriffe auch deshalb über die Zeit abfallen oder steigen, weil sich die Nutzergruppen und auch deren Verhalten ändert. War das Internet im Jahre 2004 besonders an Universitäten verbreitet, während es nun von allen Bevölkerungsgruppen genutzt wird, könnte dies einen Abschwung an Begriffen wie z.B. „Wissenschaft“ oder „Humboldt-Universität“ erklären (Stephens-Devidowitz and Varian, 2015).

Um die Häufigkeit eines Suchbegriffes bei Google als Daten zu erfassen, gibt es zwei Methoden. Google selbst bietet die Möglichkeit bis zu fünf verwendete Suchbegriffe als CSV-Datei zu exportieren. Eine zweite Möglichkeit ist extern z.B. mit Hilfe der Software *R* und den beiden Paketen *googletrend* und *okugami79*. Hierbei wird jeweils ein Suchbegriff ausgegeben, welcher für den voreingestellten Zeitraum und Ort als Tabelle und grafische Zeitreihe in *R* sowie als CSV-Datei gespeichert wird. Diese einzelnen Tabellen können, wenn mehrere Suchbegriffe analysiert werden sollen, zusammengefügt und als Datenbank gespeichert werden. Neben den beiden oben genannten Filtern bietet dieses Verfahren zudem die Möglichkeit, die Datenpunkte sowohl in monatlicher als auch wöchentlicher Periodizität auszugeben (okugami79, 2013). Abbildung A.9 zeigt die deskriptive Statistik der deutschen Wörter für Deutschland und Abbildung A.10 für die Vereinigten Staaten.

Der Vollständigkeit halber sollte noch eine dritte Methode genannt werden, mit der es möglich ist passende Suchbegriffe zu generieren. Da mit den verwendeten Daten immer ein Bezug zu abhängigen Variablen (Konjunkturdaten) hergestellt werden soll, stellt Google mit seinem Service Google Correlate (Mohebbi et al., 2011) eine Möglichkeit bereit, Zeitreihen als Input zu verwenden und korrelierende Suchbegriffe als Output auszugeben.

Hierbei wird eine Zeitreihe im CSV Format hochgeladen. Das Ergebnis sind die 100 am höchsten korrelierenden Suchbegriffe über die Zeit. Weiter ist es möglich, anstatt einer Zeitreihe ein Wort einzugeben und ebenso den gleichen Output zu bekommen. Ersteres Verfahren funktioniert aktuell jedoch nicht. Beim hochladen einer Datei meldet Google einen Fehler (Stand 01.06. - 15.08.2015). Die Worteingabe führt jedoch zu den gewünschten Ergebnissen, welche als CSV-Datei exportiert werden können.

Um zu zeigen, dass es durchaus sinnvolle Zusammenhänge zwischen der Häufigkeit gewisser Suchbegriffe und Konjunkturindikatoren gibt, wurde im Folgenden eine Panel Datenreihe erstellt. Es kann gezeigt werden, dass es einen Trend in der Frequenz, mit welcher in Deutschland nach dem Begriff „Arbeitsamt“ gesucht wird, und der Arbeitslosenquote gibt. Zudem besteht ein Zusammenhang zwischen den einzelnen Bundesländern. Dazu wurden mit Google Trends die Häufigkeiten des Suchbegriffs Arbeitsamts pro Jahr und je Bundesland ausgewertet. Ebenso ist vom Statistischen Bundesamt die Arbeitslosenquote in dieser Aufteilung verfügbar. Die Abbildung 3.1 zeigt, dass beide Datensätze mit der Zeit sowohl in ihrer Amplitude abnehmen, als auch, dass diejenigen Bundesländer, welche hier jeweils nach Höhe der Quote und relativer Häufigkeit angeordnet sind, oft den gleichen Rang belegen. In den Jahren 2005 bis 2008 hatte das Bundesland Mecklenburg-Vorpommern in beiden Datensätzen sowohl die höchste Arbeitslosenquote als auch das höchste Aufkommen an gemittelten Suchbegriffen nach „Arbeitsamt“. Auch ist in diesem Zeitraum in 75% der Fälle Sachsen-Anhalt auf dem vorletzten Platz. Über den gesamten Zeitraum teilen sich die Bundesländer Baden Württemberg, Bayern und Rheinland-Pfalz die ersten drei Plätze in der Höhe der Arbeitslosigkeit. Dies kann für die Google Daten im Zeitraum 2012 bis 2014 ebenfalls bestätigt werden. Um zu überprüfen, ob es einen saisonalen Effekt in den Suchanfragen gibt, wurden für die Arbeitslosenquote im Jahr 2015 lediglich Daten bis April erhoben. Die Wintermonate zeigen somit einen evtl. fälschlichen Anstieg im Vergleich zum Vorjahr. Dieser Effekt zeigt sich jedoch nicht in einer höheren Frequenz des Wortes „Arbeitsamt“. Es kann vermutet werden, dass jene Internetnutzer, welche von saisonaler Arbeitslosigkeit betroffen sind, nicht jedes Jahr aufs neue nach ihrem zuständigen Arbeitsamt suchen müssen. Weiter könnte man daraus folgern, dass die Google Daten, wenn es denn einen signifikanten Ausschlag gibt, eher strukturelle und/oder konjunkturelle Veränderungen abbilden.

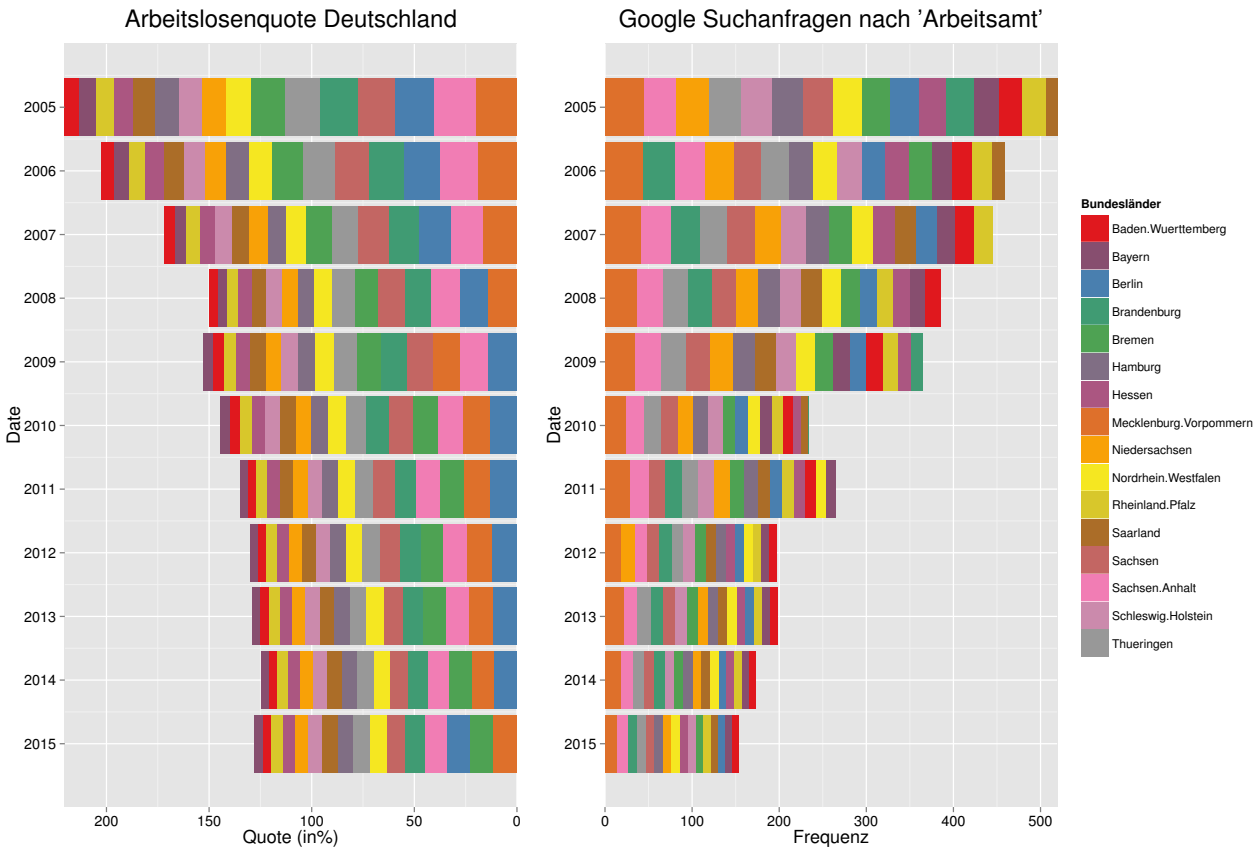

Abbildung 3.1: Arbeitslosenquote in Deutschland nach Bundesland laut Statistischem Bundesamt (links) und Suchbegriff „Arbeitsamt“ nach Bundesland laut Google Trends (rechts). Quelle: Eigene Darstellung, Daten: Statistisches Bundesamt und Google Trends.

4 Methodik und Analyse

In diesem Kapitel wird beschrieben, welche Vorarbeit auf große Datenmengen anfällt und wie diese zur weiteren Nutzung umgeformt werden können. Danach werden Methoden vorgestellt, mit denen die These, dass Big Data als Frühsignale für Konjunkturindikatoren verwendet werden können, überprüft werden kann. Sowohl eine theoretische Beschreibung der Methoden als auch eine empirische Anwendung auf einige Modelle wird in diesem Abschnitt behandelt. Die Anwendung der Vorbearbeitung von Daten und der darauf aufbauenden Methoden wird komplett mit *R* umgesetzt. Der dazugehörige Quellcode wird in Form von Quantlets digital zur Verfügung gestellt (Borke and Härdle, 2015). Die ausgewählten Verfahren befinden sich jeweils in einem eigenen Ordner, in welchem der Code und die dazugehörigen Datensätze zu finden sind, auf Quantnet.

4.1 Bearbeitung von Big Data - Dimensionsreduktion

Um mit großen Datenmengen (z. B. viele Dokumente mit langen Texten) zu arbeiten, diese also auf statistische Regressionsmodelle und andere Methoden anzuwenden, müssen diese in eine organisierte Form gebracht werden. Beim *Text Mining* geht es dabei immer um eine Dimensionsreduktion ohne dabei viel Informationsgehalt zu verlieren. Dadurch lassen sich Input-Variablen nicht nur statistisch einfacher auswerten, auch die intuitive Interpretation ist bei wenigen Dimensionen einfacher. Die Abbildung 4.1 zeigt eine Übersicht von 300 Wörtern, welche mindestens 50-mal in den Nasdaq Texten vorkommen.

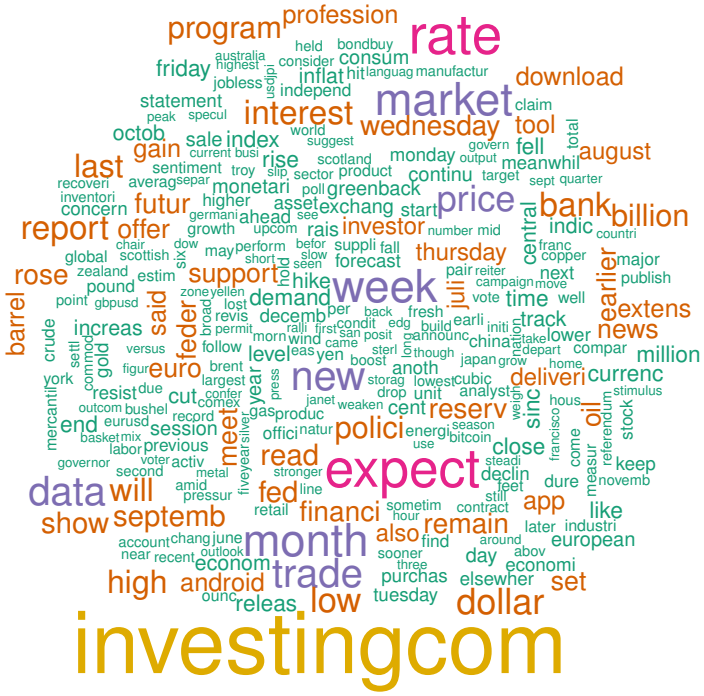

Abbildung 4.1: WordCloud der Nasdaq-Daten. Maximal 300 Wörter und Mindestfrequenz von 50. Quelle: Eigene Darstellung, Daten: Nasdaq Datensatz (Stichprobe von 300 Texten der „investing“ Artikel).

4.1.1 Latent Semantic Analysis - LSA

Der Latent Semantic Analysis (LSA) geht das Latent Semantic Indexing (LSI) voraus. In letzterem werden große Textmengen indiziert und dadurch eine Dimensionsreduktion durchgeführt. Dabei ist diese Methode unabhängig von subjektiven Entscheidungen wie vordefinierten Wörterbüchern, der Grammatik, semantischen Netzwerken oder anderem

(Landauer et al., 1998). Die Vorgehensweise kann laut Geiß and Klein-Bering (2003) in folgende Schritte unterteilt werden.

1. Erstellen einer Liste aller im Dokument vorkommenden Wörter.
2. Entfernen der Stoppwörter: Für die Analyse irrelevante Wörter sind zum Beispiel „und“, „als“ usw. aber auch jene, welche in einem Dokument nur einmal vorkommen und somit für eine Analyse nicht verwendet werden können.
3. Erstellen einer Term-Dokument-Matrix (TDM). Die relevanten Inhaltswörter werden in eine Matrix geschrieben.
4. Gewichtung der Wörter lokal (Wortwichtigkeit) und global (Informationsgehalt im Dokument).
5. Singulärwertzerlegung (*Singular Value Decomposition, SVD*): Aufspaltung der TDM in drei neue Matrizen zur Dimensionsreduktion der Ausgangsmatrix.

Bei der Singulärwertzerlegung wird die TDM A in drei Matrizen T, S und D umgeformt. In T stehen die linken Eigenvektoren (transformierte Darstellung der Terme), in S die Singulärwerte von A und in D die rechten Eigenvektoren (transformierte Darstellung der Dokumente).

$$A_{t\times d} = T_{t\times m} S_{m\times m} (D^T)_{m\times d} \tag{4.1}$$

$$\hat{A}_{t\times d} = \hat{T}_{t\times k} \hat{S}_{k\times k} (\hat{D}^T)_{k\times d} \tag{4.2}$$

Abbildung 4.2 zeigt die Umformung der Ausgangsmatrix A. Die Spalten der Matrix T zeigen, welchen Wert die Singulärwerte annehmen. Spalte 1 weißt durchgängig negative Werte (blaue Färbung) auf. Die bedeutet einen positiven Zusammenhang zwischen den Termen und der Dokumentenstruktur. Die weiteren Hauptkomponenten zeigen immer mehr eine sowohl positive als auch negative Verteilung der Termfrequenzen. Diese tragen dadurch immer weniger zur Approximation der Ausgangsmatrix bei. In der Matrix S sind alle Werte außer der Diagonalen gleich Null (blaue Färbung).

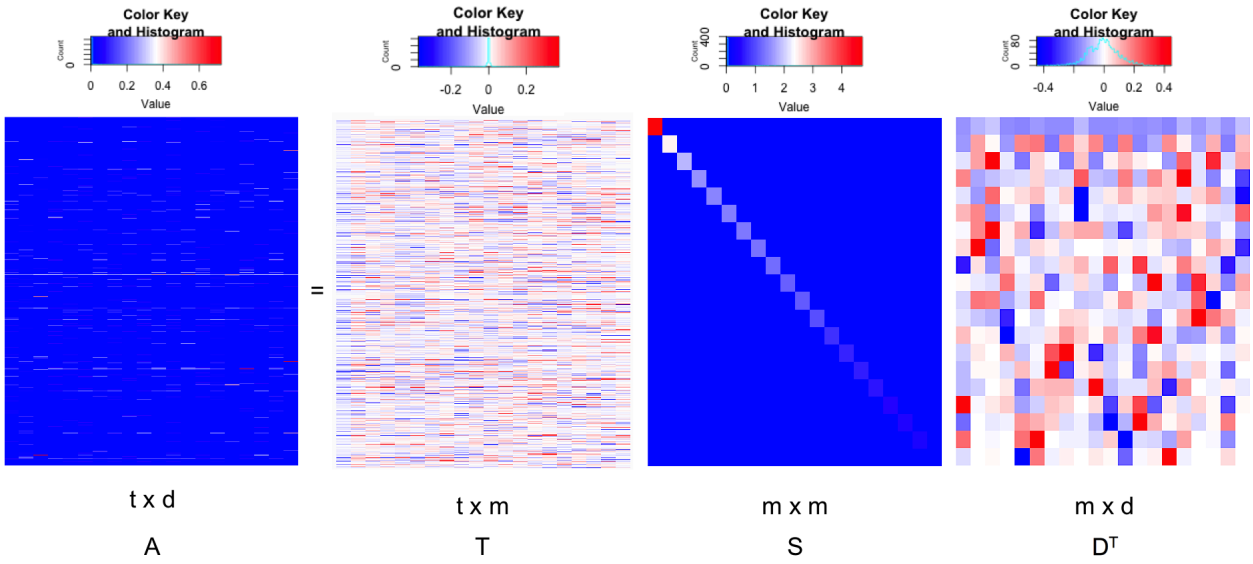

Abbildung 4.2: Singular Value Decomposition. Umformung der Originalmatrix A in drei neue. Quelle: Eigene Darstellung. Daten: Nasdaq News Texte.

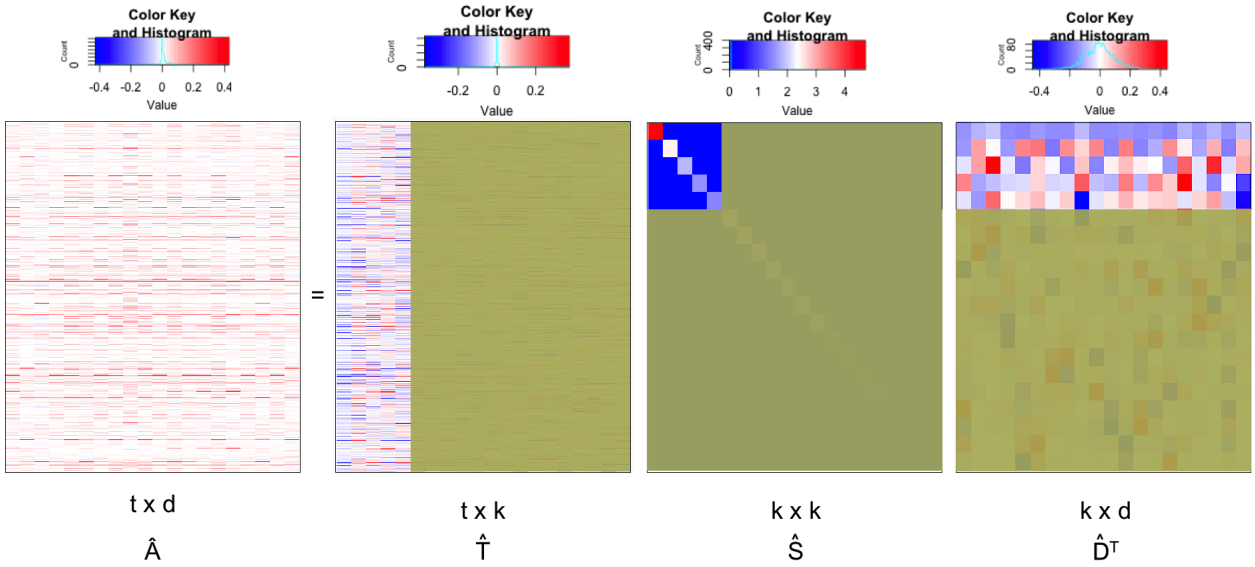

Abbildung 4.3: Singular Value Decomposition. Dimensionsreduktion auf k Singulärwerte. Quelle: Eigene Darstellung. Daten: Nasdaq News Texte.

Im nächsten Schritt werden nur die k ersten Singulärwerte berechnet und alle anderen verworfen. Dieser Schwellenwert muss empirisch ermittelt werden. Da die Elemente in den Hauptdiagonalen der Größe nach absteigend geordnet sind (Siehe Abbildung 4.2, Matrix S von rot nach blau), bleiben die wesentlichen Eigenschaften erhalten. Der Approximationsfehler ist umso kleiner, je mehr Dimensionen erhalten bleiben. Die Singulärwerte dienen dabei als Abschätzung für den Fehler. Wird nun also ein hohes Gewicht an Singulärwerten

beibehalten, können große Fehler vermieden werden. Das Ergebnis ist eine optimale Anzahl an Dimensionen. Dies wird laut Lindh-Knuutila (2014) als *truncated SVD* bezeichnet (Siehe Formel 4.2).

Abbildung 4.2 zeigt diese erste Umwandlung nach Formel 4.1 und Abbildung 4.3 die Dimensionsreduktion nach Formel 4.2 graphisch. Hierbei wurden die Matrizen T, S und D jeweils reduziert. Die Ausgangsmatrix A bleibt dabei in ihrer Dimension voll erhalten.

4.1.2 Principal Component Analysis - PCA

Nach Pearson (1901) stellt eine Hauptkomponente (*Principal Component, PC*) in einem Regressionsmodell jene Linie dar, bei welcher die quadratischen Abweichungen der Datenpunkte am geringsten sind. Auf dieser Annahme aufbauend können mit der Hauptkomponentenanalyse (PCA), speziell für große Datenmengen, mehrere solcher Cluster gefunden werden, um die Dimensionen für hochdimensionale Daten zu reduzieren. Dabei werden die ursprünglichen, am besten hoch korrelierenden, Variablen durch eine orthogonale Transformation zu einer neuen Menge an Variablen geformt. Diese Hauptkomponenten sind Linearkombinationen der ursprünglichen Variablen und nach ihrem Grad der Varianz absteigend angeordnet (Trendafilov, 2014). Als rechnerische Grundlage dient die Singulärwertzerlegung (Siehe Abschnitt LSA). Daraus lassen sich bestimmte Eigenschaften einer Matrix als Produkt spezieller Matrizen ablesen (Stewart, 1993).

Als ersten Schritt wird, anders als in der Faktorenanalyse, ein niedrigdimensionaler linearer Unterraum gesucht, welcher den Datensatz am genauesten beschreibt. Auch ist die Anordnung der Vektoren, deren Rangfolge, durch die abnehmenden Eigenwerte der Kovarianzmatrix gegeben. In der PCA enthält die erste Hauptkomponente den größten Anteil der Gesamtstreuung im Datensatz. Die Variablen zwischen diesen Komponenten sind dann im Idealfall unkorreliert. Diese genaue Trennung ist laut Trendafilov (2014) jedoch nicht oft der Fall, weshalb eine einfache Interpretation der einzelnen Komponenten schwierig ist. Jede PC stellt immer noch eine Linearkombination aller originalen Variablen dar. Um diesem Problem zu begegnen können die hinteren Komponenten, welche nur eine geringe Streuung erklären gleich Null gesetzt werden. Neben dem Ziel möglichst viel Varianz zu erklären (durch Explorationsverfahren) steht die Einfachheit des Modells dem gegenüber. Dies wird durch ein Rotationsverfahren gelöst. Hier werden die Faktoren auf die Daten gedreht, sodass nur noch Faktoren übrig bleiben, auf welche die Variablen stark korrelieren (eine hohe Ladung haben). Als Gütekriterium wird hier die Summe

der Ladungsquadrate verwendet (Eigenwerte der Ladungsmatrix). Ein orthogonales Rotationsverfahren, welches für Hauptkomponentenanalysen verwendet wird ist VARIMAX und dient der inhaltlichen Interpretationshilfe. Die aufgeklärte Varianz wird dabei nicht erhöht. Trendafilov (2014) wendet diese Rotationsmethode auf einen Datensatz an und findet dadurch bessere Ergebnisse als vor der Rotation.

Er beschreibt, dass ein Problem der PCA die lineare Abhängigkeit aller Variablen ist, sodass die Vektoren nur wenige Null-Einträge (*sparsity*) besitzen. Dadurch wird die Interpretation erschwert. Eine Lösung ist das Berücksichtigen der *sparsity* als Term im Modell. Hierbei wird die generelle PCA entsprechend umgeformt. Es gibt verschiedene Möglichkeiten für diese Transformation, welche von Trendafilov (2014) unter dem Punkt 4.2 „Sparse components: definitions and algorithms“ aufgeführt wird. Dabei wird festgestellt, dass Journée et al. (2010) den effizientesten Algorithmus für sparse PCA bereitstellen. Diese verwenden sowohl ein LASSO-Verfahren als auch die Einführung eines Kardinalitätsfehlers. Ihre entwickelte generalized Power Methode (GPower) ist laut Trendafilov (2014) die wohl schnellste als auch am vielseitigsten verfügbare Methode für sparse PCA. Dieses Feld bietet jedoch noch viel Raum für weitere Forschung und Verbesserungen. In seinem Schlusswort heißt es:

> *„...fast and reliable methods for SVD/EVD do exist already for many years, but PCA, as a tool for data analysis, remains a central research topic.“* (Trendafilov, 2014)

Eine PCA kann in *R* mit der Funktion *brcomp* aus dem *stats* Package angewendet werden. Als Datensatz wurde eine zufällige Auswahl von 100 NASDAQ Texten verwendet. Vor der Analyse wurden lediglich die Stoppwörter und Zahlen aus dem Datensatz entfernt. Die dadurch entstandenen Leerräume wurden ebenfalls entfernt. Alle Großbuchstaben der verbleibenden Wörter wurden zu kleinen Buchstaben umgeformt. Das garantiert, dass ein Wort welches am Satzanfang steht gleich dem Wort ist, welches mitten im Satz aufkommt und als gleich behandelt wird. Danach wurde aus diesem Corpus eine Term-Dokument-Matrix geformt, welche die Worthäufigkeiten pro Textdokument dokumentiert. Der letzte Schritt ist die Anwendung der PCA. Als Ergebnis sind 67 Hauptkomponenten nötig, um 90% der Varianz in den Daten zu erklären. Die Abbildung 4.4 zeigt ein *biplot* für zwei Hauptkomponenten (PC1 und PC2). In diesem stellen die Sterne (schwarz) jeweils den Punktwert der Beobachtungen (Dokumente) auf der PC dar. Sterne, welche nahe aneinan-

der liegen, gehören zu Texten, welche ähnliche Punktwerte auf den Komponenten haben. Die Vektoren beschreiben die Koeffizienten der Variablen (Wörter) auf den Hauptkomponenten. Ein Vektor zeigt dabei in die Richtung, in der die Variable am ehesten durch den Vektor erklärt wird. Hier werden nur solche Wörter angezeigt, welche vom Betrag her zu den zehn größten in jeder PC gehören. Die Anzahl der Dokumente beschränkt sich auf 300. Durch die Beschränkung der Größe der Termen, kann eine Visualisierung übersichtlich dargestellt werden. Abbildung A.6 zeigt diesen Plot ohne Beschränkung.

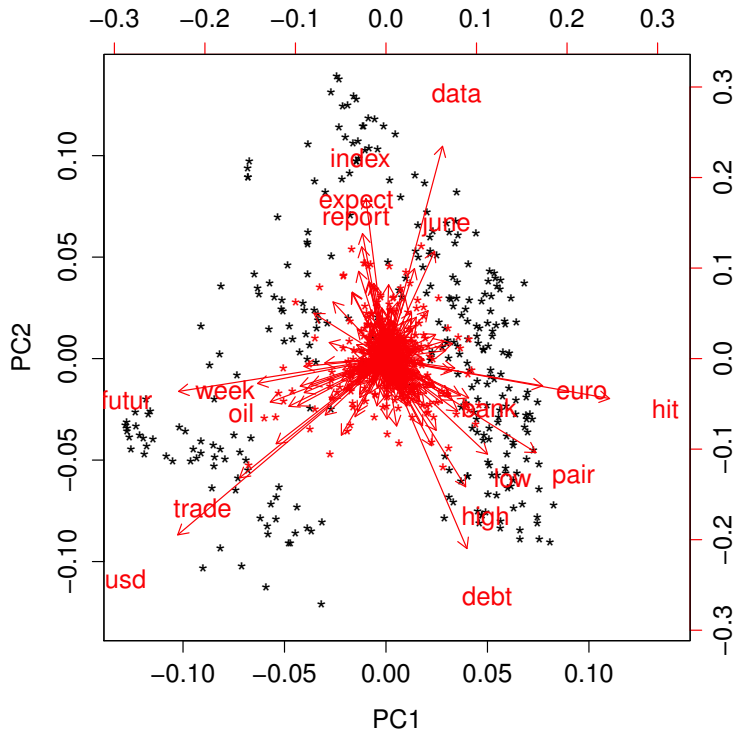

Abbildung 4.4: Biplot von zwei Hauptkomponenten. Jeweils nur die zehn betragsmäßig größten Terme pro PC. Quelle: Eigene Darstellung, Quellcode: Lukas Borke, Daten: Nasdaq Datensatz (Stichprobe von 300 Texten der „investing“ Artikel.)

4.1.3 Clustering

Text Clustering, als eine Technik des *Text Minings*, dient dazu viele und große Textdokumente in signifikante Cluster zu unterteilen. Dadurch entsteht eine organisierte Form dieser Dokumente, welche für eine weitere Anwendung sinnvoll und notwendig ist. Das Ziel ist, eine möglichst kleine intra-cluster Distanz zwischen den Dokumenten und gleichzeitig eine große inter-cluster Distanz zu erzeugen. Dazu können Ähnlichkeitsfunktionen

verwendet werden. Diese bestimmen den Grad der Ähnlichkeit sowohl zwischen zwei Textpaaren, Dokumenten, zwei Suchanfragen als auch zwischen einem Dokument und einem Begriff. Die Messfunktion nimmt dabei eine Zahl zwischen 0 und 1 an, wobei Null eine völlige Unähnlichkeit repräsentiert (Grace and Desikan, 2015). Eine oft genutzte Näherungsfunktion zum Messen der Distanz in vektorbasierten Dokumenten ist laut Grace and Desikan (2015) die euklidische Distanz

$$D_{ij} = \left(\sum_{k=1}^{d} |x_{ik} - x_{jk}|^2 \right)^{1/2} \tag{4.3}$$

wobei d die Dimensionalität des Datenobjekts und x_{ik} und x_{jk} jeweils die k-te Komponente der i-ten und j-ten Objekte von x_i und x_j darstellen. D_{ij} ist die Approximation zwischen x_i und x_j.

Zum Clustern solcher Textdokumente stehen verschiedene Verfahren zur Verfügung, welche im Folgenden kurz erläutert werden.

Das *Partitioning Clustering* formt mit seinem Algorithmus die Objekte n in k Partitionen, wobei jede Partition ein Cluster darstellt. Als Entscheidungskriterium wird eine Unähnlichkeitsfunktion, basierend auf der Distanz der Objekte, angewendet. Dadurch weisen die Objekte in den Clustern starke Ähnlichkeiten und zwischen den Clustern eine Verschiedenartigkeit auf. Methoden zur Anwendung des Partitioning Clustering sind *k-means* oder auch *k-medoids*. Unter Verwendung der Nasdaq-Daten, speziell der Artikel in der Kategorie „investing“ wird die *k-medoids* Methode angewendet. Die vorgegebene Anzahl an Clustern beträgt 8. Um die Ergebnisse graphisch darzustellen werden die Objekte durch die Projektionstechnik namens „Multidimensionale Skalierung“ (MDS) räumlich so angeordnet, dass die Distanzen zeigen, wie ähnlich sie sich sind. Dabei gilt, dass je entfernter die Objekte von einander sind, desto unähnlicher sind sie und umgekehrt (Siehe Abbildung 4.5).

Das *Hierarchische Clustering* bildet zur Gruppierung der Objekte eine Baumstruktur. Der Algorithmus kann, je nach Zerlegungsmethode, als bottom-up (dort wird zusammengeführt) oder als top-down (es wird geteilt) eingestellt werden. Bei ersterer Methode wird jedes Dokument als ein Cluster betrachtet. Danach werden jene Cluster, welche eine hohe Ähnlichkeit haben zusammengeführt. Dieser Prozess stoppt, wenn nur noch eine vordefinierte Anzahl an Cluster übrig ist (Hastie et al., 2009). Die top-down Methode verhält

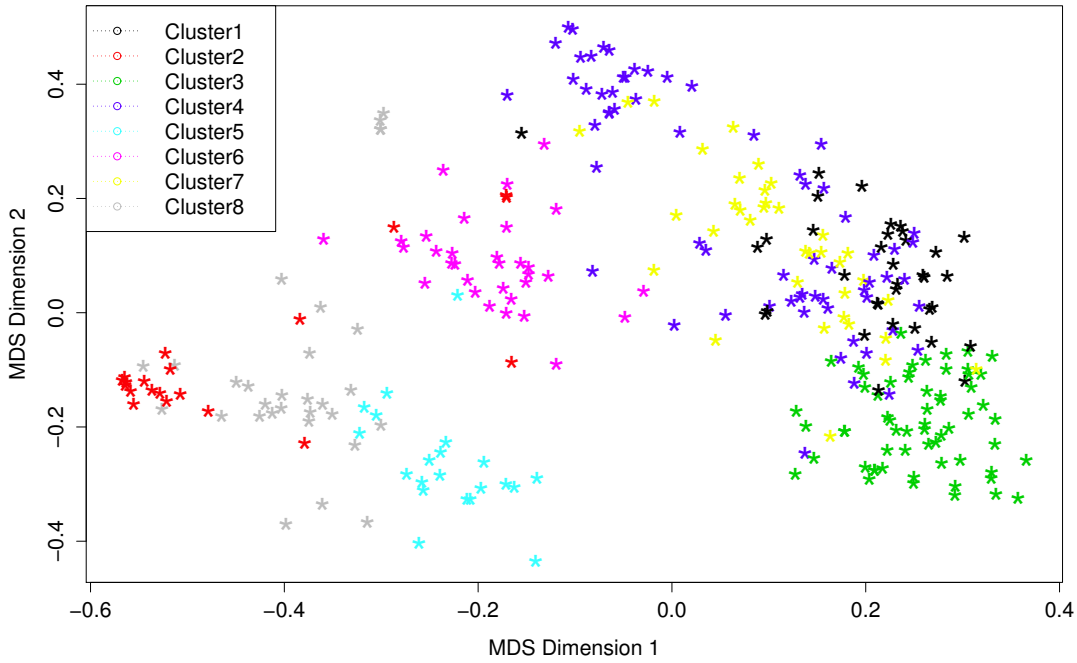

Abbildung 4.5: Multidimensionale Skalierung zur Visualisierung der Cluster. Quelle: Eigene Darstellung, Daten: Nasdaq Datensatz (Stichprobe von 300 Texten der „investing“ Artikel.)

sich genau andersrum. Hier wird im ersten Schritt genau ein Cluster gebildet welches dann immer wieder aufgeteilt wird. Hier gilt die selbe Stoppbedingung wie oben. Ein Nachteil dieser Methode ist, dass sobald eine Aufteilung (eine Gabelung im Baumsystem) vorgenommen wurde, diese nicht mehr rückgängig gemacht werden kann sollte sich herausstellen, dass diese Aufteilung in späteren Abzweigungen zu schlechten Ergebnissen führt (Grace and Desikan, 2015).

Zum auswählen von sinnvollen Clustern kann als Kriterium die Clusterqualität dienen. Dazu wird eine Funktion verwendet, welche die Dokumente und Cluster auf ein geordnetes Set von nicht-negativen reellen Zahlen zuordnet. Diese Zahlen zeigen dann wie gut ein Cluster ausgewählt wurde und somit welche Güte eine Methode besitzt. Dadurch können verschiedene angewendete Methoden verglichen werden. Eine Methode ist die Entropie (Maß für den Informationsgehalt), in der gemessen wird, wie die verschiedenen semantischen Klassen (die a priori vorgegeben oder notfalls manuell zugeordnet sein müssen) innerhalb eines Clusters verteilt wurden.

Die Entropie ist bei einem gegebenen Cluster S_r der Größe n_r wie folgt definiert:

$$E(S_r) = -\frac{1}{log\ q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} log \frac{n_r^i}{n_r} \tag{4.4}$$

wobei q die Anzahl an Klassen im Datenset beschreibt und n_r^i die Anzahl der Dokumente der i-ten Klasse welche zum r-ten Cluster gehört. Die Entropie ist so normiert, dass sie Werte zwischen 0 und 1 annimmt. Ersteres bedeutet, dass mit 100% Wahrscheinlichkeit ein konstanter Wert und keine Zufallsvariable angenommen wird. Ist die Entropie = 1 (maximale Entropie), sind alle Werte gleich verteilt. Fall „0“ tritt genau dann ein, wenn die Cluster und die a priori Klassen exakt übereinstimmen. Das Maß für den Informationsgehalt für die gesamte Clusterlösung ist die Summe der individuellen Cluster Entropien gewichtet nach der Clustergröße:

$$Entropy = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r) \tag{4.5}$$

wobei n die gesamte Anzahl an Dokumenten bezeichnet. Je kleiner die Entropiewerte, desto besser ist die Clusterlöung (Grace and Desikan, 2015).

Ein Beispiel für ein hierarchisches Clustern ist bei Ammann et al. (2014) zu finden. Sie benutzen eine Auswahl von 236 vermutlich ökonomisch relevanten Wörtern. Diese werden nach ihrer Häufigkeit des Vorkommens in der Zeitung „Handelsblatt“ sortiert. Ziel ist es, Prognosen über zukünftige DAX Änderung durch diese Wort-Indices zu erstellen. In einem ersten Schritt wurden durch eine schrittweise Regression, mit einem Eingangs- und Ausgangsschwellenwert von 10%, 26 Wörter als signifikant bezeichnet. Mit diesen startet das Clustering. Zu jedem dieser Wörter wird die euklidische Distanz ermittelt und nach dem oben beschriebenen bottom-up Prinzip jeweils ein Cluster erstellt. Dieser Prozess stoppt, bis alle Wörter miteinander verbunden sind. Ein Dendrogramm zeigt in Abbildung A.7 die Bildung der Cluster. Als Ergebnis stellen Ammann et al. (2014) fest, dass die in Cluster gefassten Terme „long term“ und „future“ beide einen positiven Ausblick auf die Zeit geben sowie „minus“ und „shaky“ auf Unsicherheit Einfluss haben. In einer Regressionsanalyse wird festgestellt, dass 8 Cluster die optimale Anzahl darstellt. Ihr R^2 (in-sample) auf DAX-Änderungen beträgt dabei 10.37%.

4.2 Big Data Analysen

Big Data kann verschiedene Formate, wie photographisch, binär und auch numerisch annehmen. Bei letzterem wird zwischen „tall“ (viele Beobachtungen T aber nicht so viele Variablen N (T»N)), „fat“ (nicht so viele Beobachtungen aber viele Variablen (N>T)) und „huge“ (viele Beobachtungen und viele Variablen (T>N)) unterschieden (Hendry and Doornik, 2014). Mit strukturierten Daten und einer Reduktion der möglichen Prädiktoren ist es nun möglich verschiedene statistische Methoden anzuwenden, um die Güteeigenschaften von exogenen auf bestimmte endogene Variablen zu überprüfen. Auch in diesen Methoden werden nochmals Variablen ausgewählt und andere verworfen, um eine statistische Signifikanz zu gewährleisten und eine Überbenutzung der Prädiktoren zu verhindern. Ziel ist es, mit möglichst wenig Input gute Out-of-Sample Ergebnisse zu produzieren. Klassische Modelle bewerkstelligen dies sehr gut. Somit ist es sinnvoll, dass in der *mashine learning world* die Komplexität eines Modells als Kosten berücksichtigt wird. Dies nennt man *Regularisation*. Um optimale Ergebnisse mit Big Data zu produzieren schlägt Varian (2014) vor, die Daten in Trainings-, Test- und Bewertungssets zu unterteilen. Dabei wird das Trainingsset zum Schätzen, das Bewertungsset zum Auswählen und das Testset zum Bewerten des Modells verwendet. Im Folgenden werden sowohl Methoden für Querschnittsdaten (Punkt 4.2.1) als auch für Zeitreihen (Punkt 4.2.2 und 4.2.3) vorgestellt. Die lineare Regression und das Bayesian Structural Time Series Verfahren werden dann mit Google Trends Daten empirisch angewendet und analysiert.

4.2.1 Baum-Modell

Von den bewährten Methoden für Entscheidungsmodelle wie z.B. LOGIT oder auch PROBIT, welche lineare Modelle sind, unterscheidet sich ein Baum-Modell (*classification and regression trees, CART*) weniger in der Form als viel mehr in der Entscheidung am Ende. Hierbei geht es um das Erstellen und Erweitern von Entscheidungsebenen welche gute Out-of-Sample Ergebnisse produzieren. Baum-Modelle können sowohl zur Klassifikation als auch zur Regression verwendet werden (Varian, 2014).

Der Aufbau dieses Modells wird von Hastie et al. (2009) wie folgt beschrieben. Durch zwei Input-Variablen (X1 und X2) sollen verschiedene Werte einer Output-Variable Y beschrieben werden. Wie Abbildung 4.6 zeigt, ist es in der linken Abbildung schwierig, für alle Bereiche durch X1 und X2 genaue Werte für Y zu definieren. Deshalb werden,

wie auf der rechten Seite zu sehen, die Bereiche binär aufgeteilt. Dazu wird der Bereich zuerst in zwei Regionen unterteilt und darauf die erhaltenen Durchschnittswerte von Y modelliert. Dann werden die Variable und der Punkt ausgewählt, bei denen der Bereich geteilt wurde, um eine beste Schätzung zu bekommen. Danach werden eine oder auch beide Regionen weiter unterteilt. Dieser Prozess wiederholt sich bis ein voreingestellter Stoppwert erreicht wird.

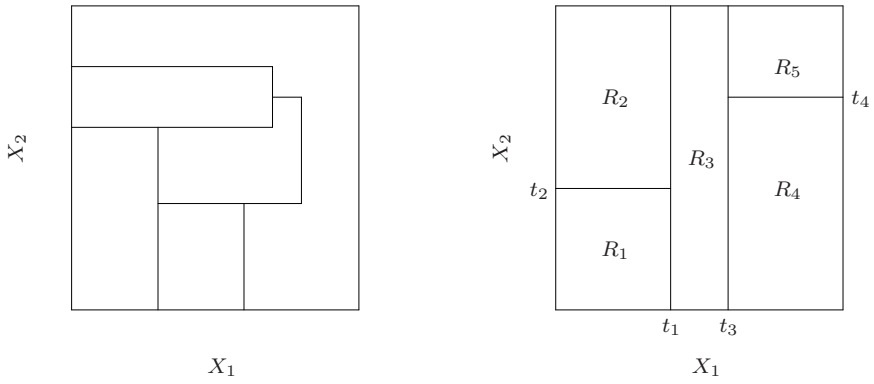

Abbildung 4.6: Regressionsfunktion (links) und binäre Unterteilung (rechts). Übernommen von: (Hastie et al., 2009).

In diesem Beispiel wurden fünf Bereiche (R1 bis R5) erstellt. Das korrespondierende Regressionsmodell schätzt Y mit einer Konstanten c_m in der Region R_m, sodass

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\left\{(X_1, X_2) \in R_m\right\}. \tag{4.6}$$

Als Baumstruktur wird das Modell in Abbildung 4.7 dargestellt. Hier werden die binären Entscheidungen jeweils an einer Kreuzung beschrieben und somit je nach Werten der Input-Variablen die zugehörigen Regionen, in welchen das vermutete Y liegt, ausgewählt.

Im Folgenden wird beschrieben, wie sich laut Hastie et al. (2009) aus dieser Klassifikation ein Regressionsbaum erstellen lässt. Es gibt p Input-Variablen und jeweils eine zugehörige Output-Variable ((x_i, y_i) für $i = 1, 2, ...N$, mit $x_i = (x_{i1}, x_{i2}, ...x_{ip})$) für jede der N Beobachtungen. Angenommen wird eine Unterteilung in M Regionen $(R_1, R_2, ...R_M)$. Modelliert wird das Ergebnis der Unterteilung als eine Konstante c_m für jede Region.

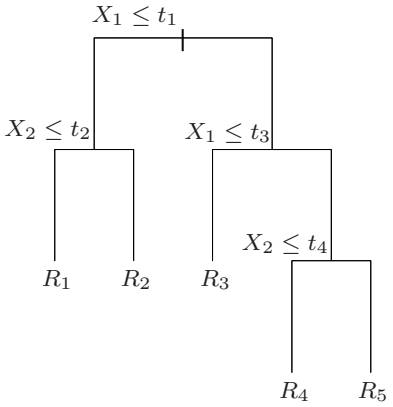

Abbildung 4.7: Baumstruktur des Regressionsmodells. Übernommen von: (Hastie et al., 2009).

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m). \tag{4.7}$$

Der beste Schätzer für $\hat{c}_m$ ist nach der kleinsten Quadrate-Methode mit $\sum(y_i - f(x_i))^2$ genau der Mittelwert von y_i in der Region R_m.

$$\hat{c}_m = ave(y_i \mid x_i \in R_m). \tag{4.8}$$

Da die Regionen in diesem Modell durch binäre Eigenschaften voneinander getrennt werden sollen, muss das Minimierungsproblem umgeschrieben werden. Zur Auswahl einer Region nehmen wir eine Trennvariable j an, welche zu Trennpunkten s führt. Dadurch entstehen im ersten Durchlauf des Modells genau zwei Regionen, beschrieben in Formel 4.9. Gesucht wird diejenige Variable j und der dazugehörige Punkt s, sodass die Summe der Abweichungen am geringsten ist. Dies lässt sich in Formel 4.10 darstellen.

$$R_1(j,s) = \{X \mid X_j \leq s\} \text{ und } R_2(j,s) = \{X \mid X_j > s\}. \tag{4.9}$$

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 \; + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]. \tag{4.10}$$

Für jede Auswahl von j und s ist der beste Schätzer $\hat{c}_i$ gleich dem Mittelwert von y_i gegeben der Werte von x_i.

$$\hat{c}_1 = ave(y_i \mid \; x_i \in R_1(j,s)) \; und \; \hat{c}_2 = ave(y_i \mid \; x_i \in R_2(j,s)). \tag{4.11}$$

Wenn unter dieser Annahme die beste Aufteilung der Regionen gefunden wurde, wird der Prozess wiederholt und jede der zwei bestehenden Regionen weiter unterteilt. Dadurch können genauere Ergebnisse erreicht werden, jedoch wird das Modell auch komplexer. Um diesem Problem zu begegnen, können Kosten für die Komplexität definiert werden. Dazu wird zunächst ein großer Baum T_0 aufgebaut, welcher erst dann stoppt, wenn z.B. nur noch fünf Datenpunkte pro Region übrig sind (Liaw and Wiener, 2002). Dieser Parameter kann, je nachdem wie groß die Datenmengen sind, variiert werden. Danach wird eine Beschneidung (*pruning*) bezüglich der Komplexität vorgenommen. Dieses Verfahren wird laut Hastie et al. (2009) als „*cost-complexity pruning*“ bezeichnet. Dabei wird ein Unterbaum T, welcher eine Teilmenge von T_0 ist, definiert. Die Endknoten werden mit m indiziert und repräsentieren jeweils die Region R_m. Weiter ist $|T|$ die Anzahl an Endknoten in T. Dadurch ergibt sich, gegeben dass N_m gleich der Anzahl der Elemente $x_i \in R_m$ ist,

$$\begin{aligned} \hat{c}_m &= \frac{1}{N_m} \sum_{x_i \in R_m} y_i, \\ Q_m(T) &= \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2, \end{aligned} \tag{4.12}$$

wodurch das Komplexitätskriterium wie folgt dargestellt werden kann

$$C_a(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|. \tag{4.13}$$

Ziel ist es, für jedes α einen Unterbaum $T_a \subseteq T_0$ zu finden, welcher das Komplexitätskriterium minimiert. Dabei steuert $\alpha \geq 0$ den Kompromiss zwischen Komplexität und Gütequalität. Große Werte für α erzeugen kleinere Bäume, wohingegen für $\alpha = 0$ der volle Baum stehen gelassen wird. Abbildung 4.8 zeigt den Zusammenhang zwischen Komplexität, welche durch die Anzahl an Abzweigungen beschrieben wird, und den Entscheidungskosten für ein unabhängiges Test-Datenset und die echten Trainingsdaten (Lewis, 2000). Um Baum-Modelle in *R* anzuwenden empfehlen sich die beiden Packages *rpart* (Therneau et al., 2015) und *party* (Hothorn et al., 2015).

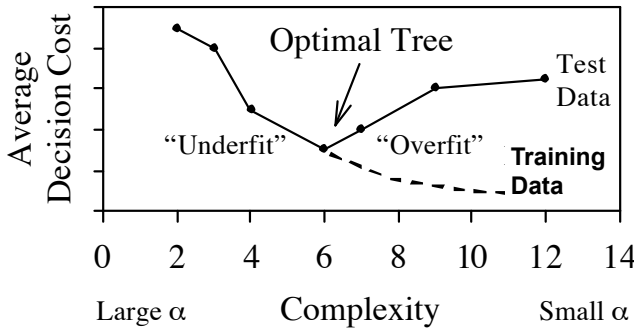

Abbildung 4.8: Optimale Baumgröße nach Komplexität im Random Forest. Quelle: (Lewis, 2000).

Eine Weiterentwicklung des einfachen Baum-Modells ist das sogenannte *Random Forest*. Es besteht aus vielen zufällig generierten Bäumen. Die Idee, beim erzeugen von multiplen Bäumen, ist die Varianzreduktion der Bagging-Methode zu verstärken. Dabei wird die Korrelation der einzelnen Bäume untereinander reduziert (*de-correlated*) ohne jedoch die Varianz maßgeblich zu erhöhen. Beim *Bagging* werden viele rauschende jedoch unverzerrte Modelle gemittelt und somit die Varianz reduziert (Hastie et al., 2009). Dem voraus geht die Bootstrap-Methode, bei der eine Stichprobe (mit Zurücklegen) der Größe n aus einem Datensatz der Größe n gezogen und die Stichprobenverteilung geschätzt wird. Wenn die beobachteten Daten $x = (x_1, x_2, ..., x_n) \rightarrow T_x$ sind, erhält man durch n-mal zufällige Ziehung mit Zurücklegen $x^* = (x_1^*, x_2^*, ..., x_n^*) \rightarrow T_x^*$. Beim *Bagging* werden viele dieser Bootstrap-Stichproben erzeugt und daraus der Mittelwert gebildet (Siehe Abbildung 4.9. Baum-Modelle eignen sich wegen ihrer tiefen Datenstruktur und der Tatsache, dass sie unverzerrt sind und ein hohes Rauschen (große Varianz) haben besonders

für diese Art der Varianzreduktion (Varian, 2014). Der Algorithmus zum *Random Forest* kann bei Foulkes (2009) und Hastie et al. (2009) nachgelesen werden.

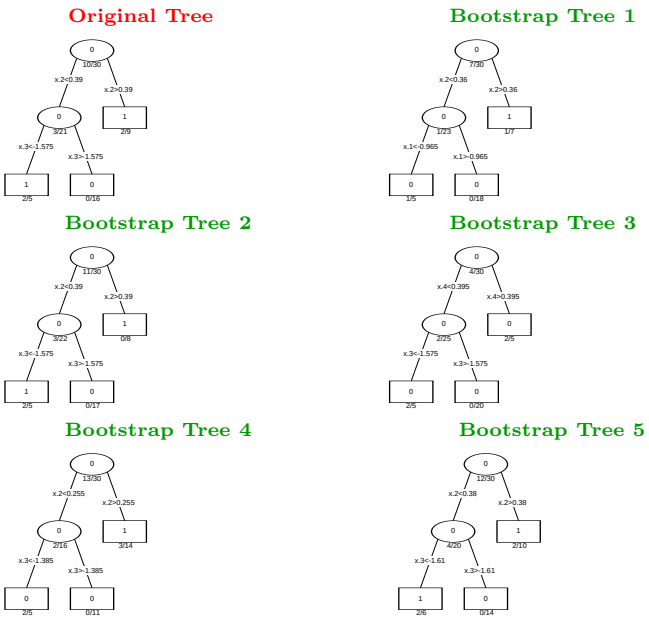

Abbildung 4.9: Bagging und Bootstrapping. Aufbau der Baumstruktur durch n-maliges Ziehen der Variablen (Bootstrap Tree). Durch Bagging entstehen mehrere solcher Bäume. Quelle: (Hastie, 2004)

Eine Implementierung von *Random Forest* in *R* kann durch das Package *RandomForests* (Liaw and Wiener, 2002) vorgenommen werden. Hierbei ist zu beachten, dass das Verfahren eine Randomisierung der Daten vornimmt. Wenn die Ergebnisse reproduziert werden sollen, empfiehlt es sich einen genauen Parameter (*seed*) für den *random number generator* festzulegen (Varian, 2014). Hastie et al. (2009) wenden drei verschiedene Methoden zum erkennen von SPAM e-Mails an. Als Ergebnis wird festgestellt, dass die *Random Forest* Methode den kleinsten Fehler produziert. Abbildung 4.10 zeigt in einer Grenzwertoptimierungskurve (*Receiver Operating Characteristic, ROC*) auf der x-Achse das Verhältnis der ausgewählten e-Mails zu den gesamten e-Mails. Die y-Achse zeigt das Verhältnis der richtigen Erkennung von SPAM (*Sensitivity*). Da wir möglichst wenig richtige e-Mails verwerfen wollen, sollte die *Specificity* bei 95% liegen. Dadurch werden bei einem einfachen Baum-Modell (blaue Linie) ca. 85% der e-Mails richtig erkannt. Unter der Verwendung von *Random Forest* (rote Linie) sind es $> 93\%$. Die grüne Linie zeigt die Verwendung des *Support Vector Mashine (SVM)* Klassifikator.

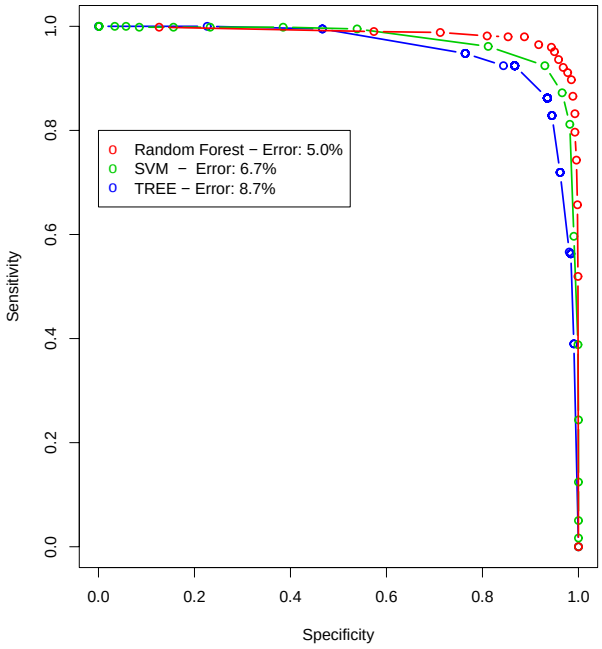

Abbildung 4.10: Güte im Baum-Modell und Random Forest. Blaue Linie zeigt den Fehler vom einfachen Baum-Modell, grün den Fehler der Support Vector Mashine und die rote Linie den Fehler unter Verwendung von Random Forest. Quelle: (Hastie, 2004)

4.2.2 Lineare Regression

Ein lineares Regressionsmodell stellt die einfachste Methode dar, um Güteeigenschaften der Input-Variablen auf die Output-Variable zu überprüfen. Hierfür werden Daten aus Google Trends als Prädiktoren (p) ausgewählt und auf verschiedene ökonomische Indikatoren (Y) regressiert. Da bei Google Trends die Anzahl an Wörtern als Suchbegriffe die möglichen zeitlichen Beobachtungen (n) übersteigt, muss hier bereits im Vorfeld eine sinnvolle Auswahl getroffen werden. Dabei kann wie folgt vorgegangen werden.

1. Bestimme manuell mögliche Prädiktoren, wobei p $<$ n als maximale Anzahl gelten soll.

2. Regressiere diese auf Y und wähle eine Anzahl x, welche die höchste Güteeigenschaft haben, aus. Als Kriterium hierfür können das adj.R^2, AIC oder auch BIC verwendet werden.

3. Suche mit den verbleibenden x Wörtern über Google Correlate die 100 am höchsten korrelierenden Suchbegriffe.

4. Regressiere nun diese neuen Suchbegriffe auf die Output-Variable.

Um eine große Anzahl an Prädiktoren zu überprüfen, kann in *R* z.B. das Package *bestglm* von McLeod and Xu (2015) verwendet werden. In diesem wird folgendes lineares Modell unterstellt:

$$y_i = \beta_0 + \beta_1 x_{i,1} + ... + \beta_p x_{i,p} + e_i \tag{4.14}$$

Wenn $p \leq 25$, kann ein effizienter sogenannter *branch and bound* Algorithmus verwendet werden, um das Modell mit dem kleinsten quadratischen Fehler der Größe m zu finden. Einzustellen ist die maximale Anzahl an verwendeten Prädiktoren und auch, ob eine Auswahl dieser fest im Modell berücksichtigt werden soll. Der Output sieht wie folgt aus:

```
1 subsets of each size up to 4
Selection Algorithm: exhaustive
         downturn economic.downturn bad.economy down.economy tough.economic.times
1  ( 1 ) " "      " "               " "          "*"           " "
2  ( 1 ) " "      " "               "*"          "*"           " "
3  ( 1 ) " "      " "               "*"          "*"           "*"
4  ( 1 ) " "      " "               "*"          "*"           "*"
         recession.proof.jobs recession.proof recession.proof.businesses hard.economic.times
1  ( 1 ) " "                  " "              " "                        " "
2  ( 1 ) " "                  " "              " "                        " "
3  ( 1 ) " "                  " "              " "                        " "
4  ( 1 ) " "                  "*"              " "                        " "
> summary(regsub)$adjr2
[1] 0.8032088 0.8179161 0.8341519 0.8369980
> summary(regsub)$bic
[1] -640.8862 -667.0464 -699.5129 -701.4716
```

Abbildung 4.11: Output des bestglm Packages. Signifikante Variablen werden mit einem Stern gekennzeichnet. Darunter findet sich das adj. Bestimmtheitsmaß und das BIC Kriterium. Quelle: bestglm in R.

Als Regressoren wurden hier Daten aus Google Trends für U.S. verwendet d.h. die wöchentliche Anzahl der Suchbegriffe im Zeitraum 07.2007 bis 07.2015 (Siehe A.10). Regressiert wurde auf die Zeitreihe des *Financial Risk Meter* vom SFB 649 welche in täglicher Periodizität vorliegt und durch Bildung der Mittelwerte auf wöchentliche Periodizität ungerechnet wurde. Die Variablen, welche im Modell den kleinsten quadratischen Fehler

ausweisen werden mit einem Stern gekennzeichnet. In Zeile 1 wird nur eine Variable (down economy) berücksichtigt. Die Anzahl steigt dann bis zur hier voreingestellten maximalen Auswahl von vier Regressoren. Verschiedene Gütekriterien können nach der Regression abgefragt werden. Hier in Abbildung 4.11 wurde das adjustierte R^2 und das BIC-Kriterium verwendet. Weitere Informationen zu einem möglichen Regressionsmodell zeigt Tabelle 4.1. Das dazugehörige Streudiagramm findet sich im Anhang unter Abbildung A.8.

Tabelle 4.1: Ergebnisse des Regressionsmodells mit einem Prädiktor (Suchbegriff: „down economy“). Regressiert auf die Lambda-Zeitreihe des Financial Risk Meters. Quelle: Eigene Abbildung aus R. FRM vom SFB649, Regressor aus Google Correlate.

	Dependent variable:
	Lambda Zeitreihe FRM
Suchbegriff: „down economy“	0.018***
	(0.0004)
Constant	0.035***
	(0.0004)
Observations	401
R^2	0.855
Adjusted R^2	0.854
Residual Std. Error	0.008 (df = 399)
F Statistic	2,346.270*** (df = 1; 399)
Note:	*p<0.1; **p<0.05; ***p<0.01

Die FRM-Zeitreihe wurde im Regressionsmodell um fünf Wochen in die Zukunft verlegt. Das heißt, dass die Daten von Google Trends heute, die Werte für den FRM in fünf Wochen abbilden. Dadurch stieg das R^2 um weitere 5% von 0.80 (zu sehen im bestglm Output) auf 0.85, wie in Tabelle 4.1 zu sehen ist.

Die Anwendung dieses Modells auf das BIP von Deutschland mit Google Trends Wörtern zeigt, dass in der begrenzten Auswahl das Wort „abschwung“ mit einem R^2 von 0,54 die höchste Güte besitzt (Siehe Tabelle B.1). Das in anderen Publikationen (Siehe z.B. (Iselin and Siliverstovs, 2013)) oft benutzte Wort „rezession“ erzeugt ein R^2 von 0.21 (Siehe Tabelle B.2).

4.2.3 Bayesian Structural Time Series (BSTS)

Die Auswahl sinnvoller Variablen ist, wenn als Prädiktoren mehrere Hundert vorhanden sind, ein Problem, um welches sich diese Methode kümmert. Nach Scott and Varian (2014a) wird ein Modell mit konstantem Level, einem linearen Zeittrend und Regressor-Komponenten angenommen:

$$y_t = \mu + bt + \beta x_t + \epsilon_t \tag{4.15}$$

Das Level μ und der Trend b sind Konstanten, x_t ist ein Vektor von Regressoren, β ein Vektor der Steigungen von x_t und ϵ_t beschreibt den Fehlerterm. Sowohl das Level als auch die Steigung folgen einer Zufallsbewegung (*random walk*), das heißt, dass sie sich über die Zeit ändern.

$$\begin{aligned}
&(1)\quad y_t = \mu_t + z_t + \epsilon_{1t} && \epsilon_{1t} \sim N(0, V) \\
&(2)\quad \mu_t = \mu_{t-1} + b_{t-1} + \epsilon_{2t} && \epsilon_{2t} \sim N(0, W1) \\
&(3)\quad b_t = b_{t-1} + \epsilon_{3t} && \epsilon_{3t} \sim N(0, W2) \\
&(4)\quad z_t = \beta x_t
\end{aligned} \tag{4.16}$$

Formel (1) spiegelt das Level und die Regression wider. (2) ist der *random walk* und der Trend, wobei (3) den *random walk* für den Trend darstellt. Die Regression wird in (4) beschrieben (Varian, 2014). Geschätzt werden sollen die unbekannten Varianzen (V, W1, W2) und die Beta-Koeffizienten. Weiter müssen sinnvolle Regressoren ausgewählt werden. Somit setzt sich nach Scott and Varian (2014a) das Modell aus folgenden Schritten zusammen:

1. Kalman-Filter, um die Zeitreihe zu bereinigen.
2. Spike and Slap Regression, um die besten Variablen auszuwählen.
3. Bayesian Model Averaging für die finale Prognose.

In Formel 4.16 ist (1) die Beobachtung wobei (2) bis (4) die einzelnen Zustände (*states*) darstellen. Im Umgang mit State Space Models kann laut DeJong and Dave (2011) der Kalman Filter und Smoother verwendet werden. Dieser errechnet rekursiv die Verteilung von $p(\alpha_{t+1}|y_{1:t})$ indem er $p(\alpha_t|y_{1:t-1})$ mit y_t in einem Standard Set an Formeln kombiniert. α beschreibt hier einen Vektor an Zustandsvariablen.

Im zweiten Schritt sollen sinnvolle Prädiktoren ausgewählt werden. Angenommen wird dabei, dass es einen Vektor γ gibt, welcher die gleiche Länge hat wie die Anzahl der möglichen Prädiktoren, welcher auch gleich der Länge von β ist. Der Vektor γ kann entweder den Wert Eins oder Null annehmen und kennzeichnet somit, ob eine Input Variable in das Modell aufgenommen wird oder nicht. Wenn $\gamma_i = 1$, dann ist $\beta_i \neq 0$ und falls $\gamma_i = 0$, ist $\beta_i = 0$. Es gilt, dass β_γ eine Teilmenge von β ist für welche $\gamma = 1$ ist und σ^2 die Varianz der Fehler aus dem Regressionmodell darstellt. Danach kann laut Scott and Varian (2014a) ein Spike and Slap Prior für die gemeinsame Verteilung von $(\beta, \gamma, \sigma^{-2})$ wie folgt dargestellt werden:

$$p(\beta, \gamma, \sigma^{-2}) = p(\beta_\gamma|\gamma, \sigma^{-2})p(\sigma^{-2}|\gamma)p(\gamma). \tag{4.17}$$

Für γ wird eine Bernoulli Verteilung für jedes i unterstellt, da angenommen wird, dass jede Variable die gleiche Wahrscheinlichkeit besitzt, um in die Regression aufgenommen zu werden, das heißt, eben nicht gleich Null zu sein. Dieser Teil wird auch als *Spike* bezeichnet (Varian, 2014).

$$\gamma \sim \prod_i \pi_i^{\gamma_i}(1 - \pi_i)^{1-\gamma_i}. \tag{4.18}$$

Dabei bezeichnet π_i die Wahrscheinlichkeit, dass eine Variable aufgenommen wird. Sollte keine genaue Information über diese Aufteilung vorliegen, kann jedes π_i die gleiche Zahl zugeordnet werden. Diese ergibt sich aus der vermuteten Anzahl an aufgenommenen Variablen k zu allen Koeffizienten K, sodass $\pi = k/K$.

Die *Slap* Komponente ist ein Prior für die Werte der aufgenommenen Prädiktoren, bedingt auf dem Wissen, welche Koeffizienten ungleich Null sind. Wenn b ein Vektor von vorausgehenden Schätzungen der Regressoren, ω^{-1} eine vorausgehende Präzisionsmatrix

und ω_γ^{-1} die Spalten und Reihen der Präzisionsmatrix beschreiben, für welche $\gamma_i = 1$ ist, dann ist ein Slap Prior gleich

$$\beta_\gamma|\gamma, \sigma^2 \sim N\left(b_\gamma, \sigma^2(\omega_\gamma^{-1})^{-1}\right), \tag{4.19}$$
$$\frac{1}{\sigma^2} \sim \Gamma\left(\frac{df}{2}, \frac{ss}{2}\right).$$

Laut Scott and Varian (2014a) ist es üblich, b = Null zu setzen.

Im letzten Schritt wird durch eine wiederholte Zufallsstichprobe die Form der A-posteriori-Verteilung für eine Voraussage von y_{t+1} bestimmt (Härdle et al., 2011). Die für jede Stichprobe erhobene Zahl wird dann für eine finale Voraussage gemittelt. Ebenso können, durch bilden des Mittelwerts für alle Regressionen pro Durchlauf, jene Prädiktoren ermittelt werden, welche eine hohe Wahrscheinlichkeit haben in der finalen Regression eingebunden zu werden. Die wiederholten Durchläufe werden mit Hilfe des Markov chain Monte Carlo Verfahren vorgenommen (Scott and Varian, 2014a).

Im Folgenden wird das *Bayesian Strutural Time Series* (BSTS) Verfahren mit Daten der Arbeitsmarktstatistik angewendet. Diese eignen sich für Deutschland sehr gut, da sie sowohl einen linearen Trend aufweisen, als auch eine Saisonkomponente beinhalten. Als Output-Variable dient die Arbeitslosenquote für Deutschland im Zeitraum 01.2005 bis 12.2014. Eine lineare Regression konnte im Vorfeld zeigen, dass der Begriff „Arbeitsamt“ eine sehr gute Korrelation auf die Arbeitslosenquote aufweist (Siehe Tabelle 4.2). Die Modellierung wird mit *R* und dem Package *bsts* (Scott, 2015)vorgenommen.

Danach wurden über Google Correlate die zu diesem Begriff am stärksten korrelierenden 100 Suchbegriffe extrahiert. Im BSTS-Modell wurde für die Arbeitslosenquote ein linearer Trend und eine Saisonalität hinzugefügt. Insgesamt wurden 4000 Durchläufe vorgenommen, um die Variablen zu schätzen. Die vermutete maximale Anzahl an Prädiktoren wurde auf fünf gesetzt. Abbildung 4.12 zeigt jene Variablen, welche eine hohe Güteeigenschaft besitzen und somit mit hoher Wahrscheinlichkeit in das Modell aufgenommen werden. Dabei beschreibt eine schwarze Färbung, dass ein negativer Zusammenhang zwischen dem Prädiktor und der Output-Variable herrscht. Für weiß gilt genau das Gegenteil. Zu sehen ist hier, dass die Variable „mini Job“ mit fast 100% im Modell aufgenommen wird

Tabelle 4.2: Ergebnisse des Regressionsmodells mit einem Prädiktor (Suchbegriff: „Arbeitsamt“). Regressiert auf die Arbeitslosenquote für Deutschland.
Quelle: Eigene Abbildung aus R. Arbeitslosenquote vom Statistischen Bundesamt, Regressor aus Google Trends.

	Dependent variable:
	Arbeitslosenquote
Suchbegriff: „Arbeitsamt“	0.288***
	(0.015)
Constant	4.989***
	(0.188)
Observations	122
R^2	0.801
Adjusted R^2	0.791
Residual Std. Error	0.873 (df = 120)
F Statistic	361.671*** (df = 1; 120)
Note:	*p<0.1; **p<0.05; ***p<0.01

und ein Anstieg signalisiert, dass die Arbeitslosenquote fallen könnte. Ebenso wie bei der Variable „richtig bewerben“.

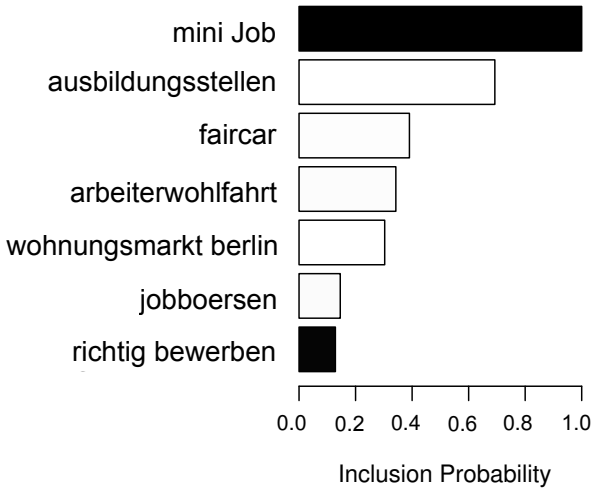

Abbildung 4.12: Wahrscheinlichkeit der Prädiktorenaufnahme in das Modell. Schwarz stellt eine negative Beziehung, weiß eine positive zur Arbeitslosenquote dar. Quelle: Eigene Abbildung aus R. Arbeitslosenquote vom Statistischen Bundesamt, Regressoren aus Google Correlate.

In der folgenden Abbildung (4.13) wird eine schrittweise Zusammensetzung des Modells gezeigt. Im ersten Schritt wird über die Zeitreihe ein lokaler linearer Trend gelegt (oben links). Danach folgt die Einbindung der Saisonalität (oben rechts). Die nun schwache rote Linie zeigt jeweils die Zeichnung des Modells im vorherigen Schritt. Allein durch die Berücksichtigung von saisonalen Schwankungen kann die Zeitreihe sehr gut abgebildet werden. Im letzten Schritt werden nun sukzessiv die einzelnen Prädiktoren hinzugefügt.

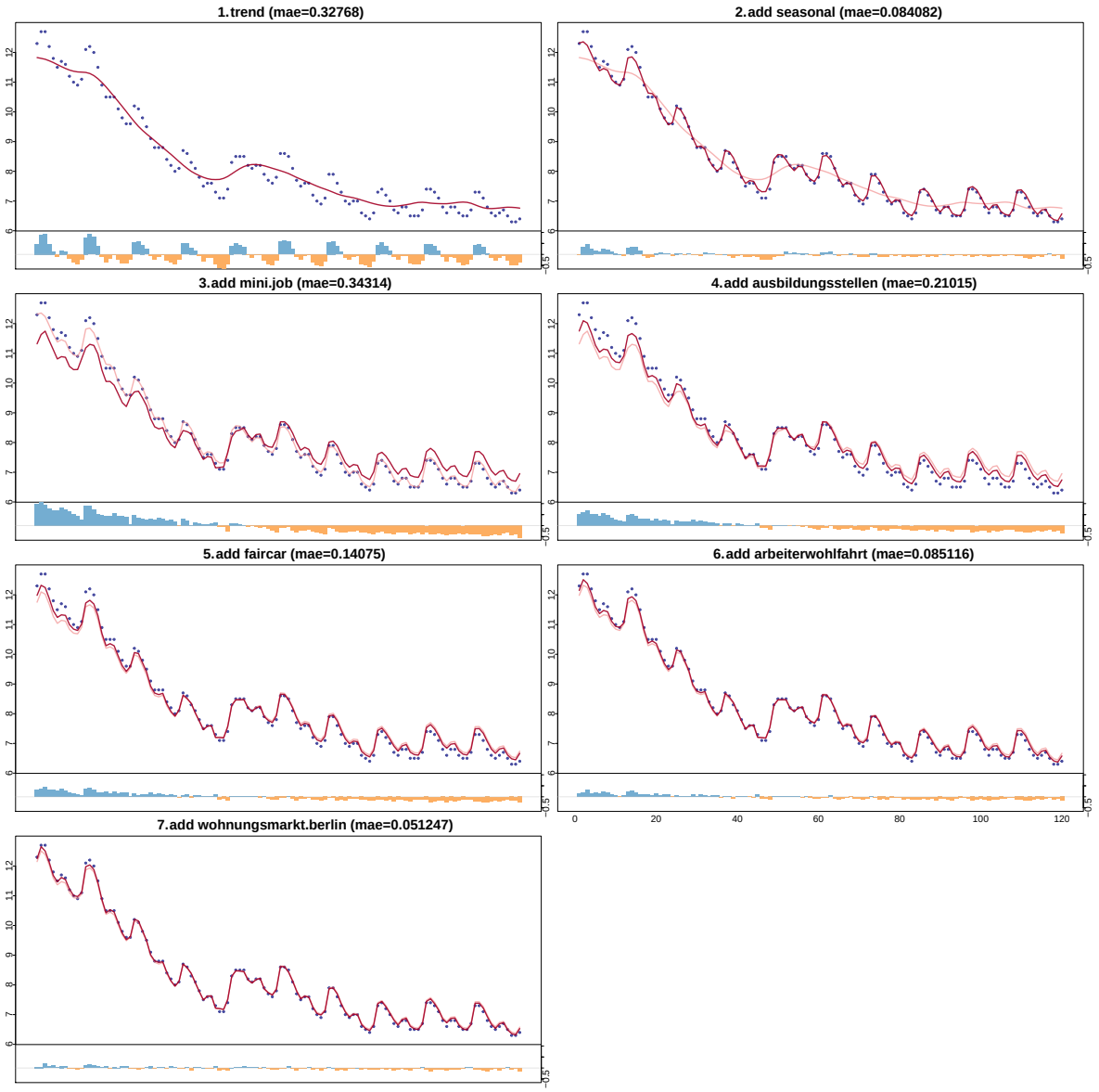

Abbildung 4.13: Schrittweise Modellzusammensetzung. Trend, Saisonkomponente und die fünf am besten passenden Regressoren. Quelle: Eigene Abbildung aus R. Daten: Arbeitslosenquote vom Statistischen Bundesamt, Regressoren aus Google Correlate, Plotfunktion von (Varian, 2014).

Eine Überprüfung der Prädiktoren im linearen Regressionsmodell ergibt für den Begriff „mini Job“ einen Anstieg im R^2 von 80% auf 81% (Siehe Tabelle 4.3). Wird nun noch der Begriff „Ausbildungsstelle“ dazugenommen steigt das Bestimmtheitsmaß auf 85% wie Tabelle 4.4 zeigt.

Die Abbildungen 4.14 und 4.15 zeigen die Anwendung der BSTS-Methode auf die Lambda Zeitreihe des *Financial Risk Meters* vom SFB 649. Die Regressoren kommen dabei ebenfalls von Google Trends und Google Correlate. Auf Grundlage der Ergebnisse der linearen Regression (Siehe Tabelle 4.1) wurde zu dem Begriff „down economy“ über Google Correlate 100 weitere hoch korrelierende Begriffe gesucht. In einem weiteren Schnitt wurde der Datensatz manuell bearbeitet. Es wurden zweifelhafte Begriffe wie „lyrics xy“ oder auch „Acai Berry“ gelöscht. Beide Zeitreihen wurden auf eine monatliche Basis durch Bildung von Mittelwerten hochgerechnet. Anders als bei der Arbeitslosenquote wurde bei der Lambda Zeitreihe zwar ein linearer Trend, jedoch keine Saisonalität unterstellt.

Tabelle 4.3: Ergebnisse des Regressionsmodells mit einem Prädiktor (Suchbegriff: „mini Job“). Regressiert auf die Arbeitslosenquote für Deutschland. Quelle: Eigene Abbildung aus R. Arbeitslosenquote vom Statistischen Bundesamt, Regressor aus Google Trends.

	Dependent variable:
	Arbeitslosenquote
Suchbegriff: „mini Job“	2.271***
	(0.099)
Constant	8.721***
	(0.072)
Observations	120
R^2	0.815
Adjusted R^2	0.814
Residual Std. Error	0.754 (df = 118)
F Statistic	521.351*** (df = 1; 118)
Note:	*p<0.1; **p<0.05; ***p<0.01

Tabelle 4.4: Ergebnisse des Regressionsmodells mit zwei Prädiktoren (Suchbegriff: „mini Job“ und „Ausbildungsstellen“). Regressiert auf die Arbeitslosenquote für Deutschland. Quelle: Eigene Abbildung aus R. Arbeitslosenquote vom Statistischen Bundesamt, Regressoren aus Google Trends.

	Dependent variable:
	Arbeitslosenquote
Suchbegriff: „mini Job“	0.426
	(0.345)
Suchbegriff: „Ausbildungsstellen“	1.569***
	(0.283)
Constant	8.598***
	(0.068)
Observations	120
R^2	0.854
Adjusted R^2	0.851
Residual Std. Error	0.674 (df = 117)
F Statistic	341.550*** (df = 2; 117)
Note:	*p<0.1; **p<0.05; ***p<0.01

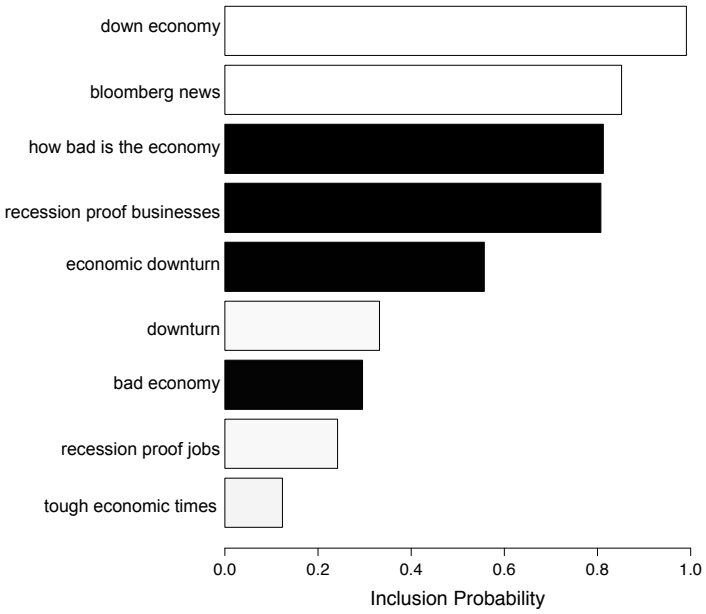

Abbildung 4.14: Wahrscheinlichkeit der Prädiktorenaufnahme in das Modell. Schwarz stellt eine negative Beziehung, weiß eine positive zum *Financial Risk Meter* (Lambda) dar. Quelle: Eigene Abbildung aus R. Financial Risk Meter vom SFB 649, Regressoren aus Google Correlate.

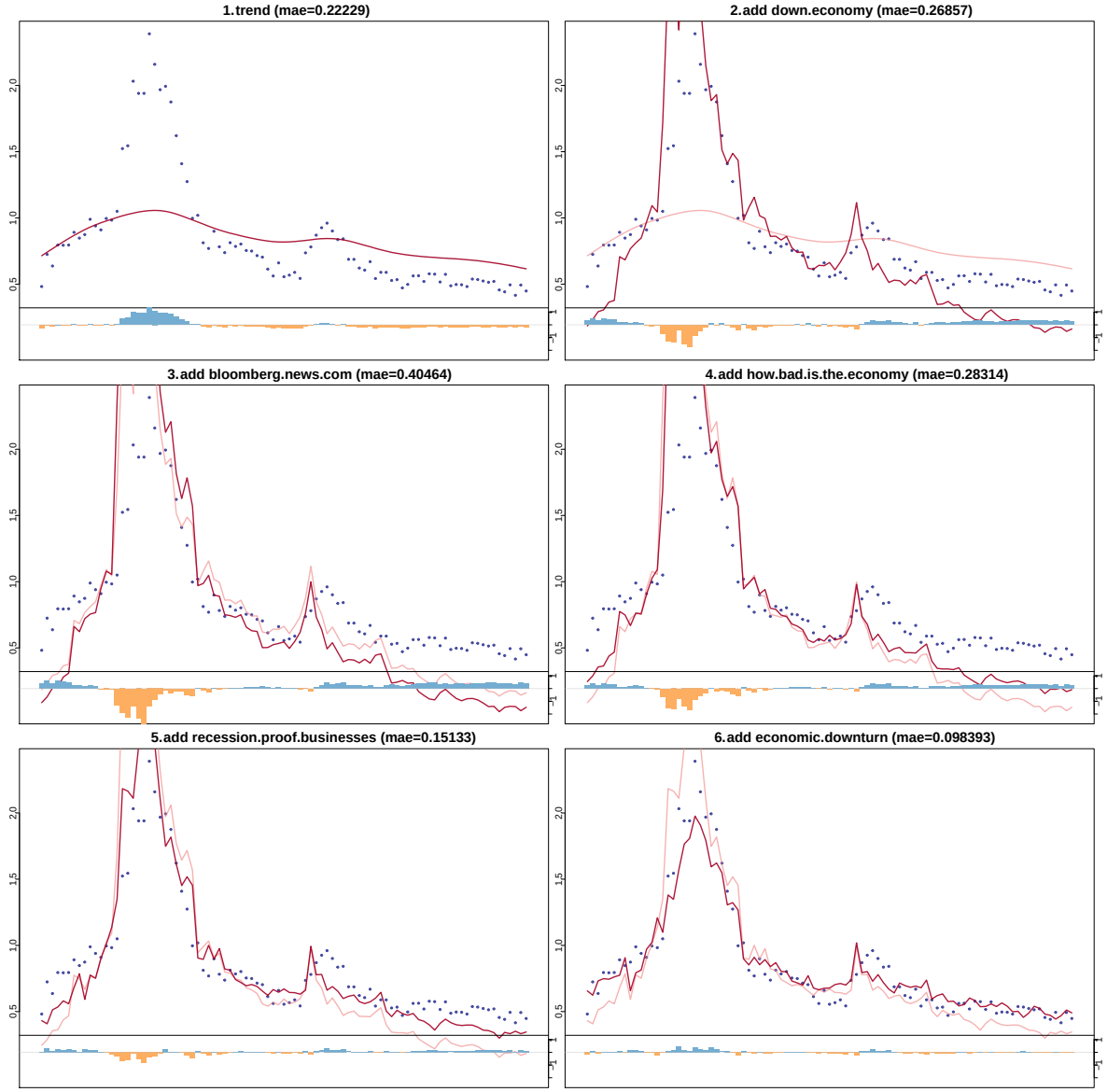

Abbildung 4.15: Schrittweise Modellzusammensetzung. Trend, Saisonkomponente und die fünf am besten passenden Regressoren. Quelle: Eigene Abbildung aus R. Daten: Financial Risk Meter vom SFB 649, Regressoren aus Google Correlate, Plotfunktion von (Varian, 2014).

5 Zusammenfassung

Das Frage nach Frühsignalen in großen Datenmengen kann nicht mit einem eindeutigen Ergebnis beantwortet werden. Diese Arbeit liefert vielmehr einen Überblick im Umgang mit diesem experimentellen Bereich des Data Minings und stellt dabei mögliche Vorgehensweisen vor. Es soll nicht der Eindruck vermittelt werden, dass durch einfaches Filtern von oft verhaltensabhängigen Daten die zukünftige Entwicklung einer Wirtschaftszone prognostiziert werden kann. Die Entwicklung des Verständnisses für den Begriff „Big Data“, dessen Potenzial aber auch damit einhergehende Probleme zu verstehen steht im Vordergrund.

Eine erste Schwierigkeit stellt die Beschaffung solcher Datensätze dar. Soziale Netzwerke wie Facebook oder Twitter haben immer stärkere Beschränkungen in ihrer Programmierschnittstelle (*Application Programming Interface, API*). Dies führt dazu, dass es mit Stand September 2015 bei Twitter nur noch möglich ist, 1% der Tweets und diese nur maximal sieben Tage in der Vergangenheit auszulesen. Die theoretischen Möglichkeiten, sich automatisiert Daten zu generieren werden jedoch zunehmend vereinfacht. So gibt es Packages für *R*, welche verschiedene soziale Netzwerke analysieren können.[1] Google hingegen stellt die kompletten Suchanfragen ab 2004 zur Verfügung. Interessant wäre auch, Texte aus Zeitungen zu analysieren. Dazu gibt es Papers (Siehe Iselin and Siliverstovs (2013); Ammann et al. (2014)), in denen die Autoren auf Texte des Handelsblatts zurückgreifen. Eine Anfrage für diesen Datensatz scheiterte an zu hohen Kosten. Anfragen bei anderen Tageszeitungen blieben erfolglos.

Die Vorbearbeitung von Datensätzen unterläuft mehrere Schritte. Dabei kann die Dimensionsreduktion durch verschiedene Methoden vorgenommen werden. Hierbei unterscheidet sich die LSA von der PCA. Bei LSA baut die SVD auf einer Term-Dokument-Matrix auf, die PCA hingegen bei einer Kovarianzmatrix. Die Ausgangsmatrix A ist somit verschieden und es werden unterschiedliche Ergebnisse produziert. Ebenso sind bei LSA die Eigenvektoren(Links-Singulärvektoren) der Matrix D Vektoren mit Term-Koordinaten (oder auch semantische Hauptkomponenten), bei PCA hingegen Hauptkomponenten (mit größten Varianzanteilen).

Das LSI ist effektiv auf Synonyme anwendbar. Probleme entstehen jedoch im Bereich der Polysemie. Das heißt, dass ein Wort mehrere Bedeutungen haben kann. Zum Beispiel

[1] Für Twitter z.B. das Package *twitteR*

kann das Wort „Apple“ zum einen die Frucht sein, also in die Kategorie Nahrungsmittel gehören oder aber auch der Name des Computerherstellers und somit zu IT/Computer zugeordnet werden (Deerwester et al., 1990; Landauer et al., 1998; Trendafilov, 2014).

Die Durchführung der PCA mit einem Sample der NASDAQ Texte ergab 68 Hauptkomponenten, um mindestens 90% der Varianz der Daten zu erklären. Da lediglich Zahlen und Stoppwörter aus den Texten entfernt wurden, ist dieser immer noch viel zu groß, um ein einfaches Modell zu entwickeln. Ähnliche Resultate ergeben sich, wenn dieser Datensatz in Cluster aufgeteilt wird. Als Ergebnis werden die mehr als 65.000 noch verbleibenden Wörter zu über 100 Cluster zusammengefasst. Auch hier ist keinerlei übersichtliche Graphik zu erstellen. Ich stimme somit Ammann et al. (2014) zu, welcher nur seine Auswahl von 236 Wörtern in den jeweiligen Texten übrig lässt. Dadurch entstehen als Ergebnis eine übersichtliche Anzahl an Clustern welche zur weiteren Analyse verwendet werden können.

Irrelevante Wörter führen zu einem weiteren Problem bei großen Datensätzen. Über Google Correlate habe ich nach Wörtern gesucht, welche evtl. hoch mit der Zeitreihe des *Financial Risk Meters* korrelieren. Als mögliches Wort, mit einem R^2 von 0.72, wird „Acai Berry“ vorgeschlagen. Dies liegt jedoch lediglich daran, dass die Nachfrage nach dieser Beere in den Jahren der „Great Recession“ unabhängig davon, sehr stark anstieg. Wenn Wörter also nicht im Vorfeld, wie oben beschrieben, selektiert werden, so muss dies in jedem Fall nachträglich geschehen. Auch Varian (2014) wendet diese Art der Nachbearbeitung auf seinen durch Google Correlate erzeugten Datensatz an, indem er z.B. das Wort „oldies lyrics“ entfernt.

Wenn nun die Datensätze dementsprechend bearbeitet wurden, können auf diese anschließend statistische Methoden angewendet werden. Das Package *bestglm* in *R* bietet dabei eine Möglichkeit, um schnell einen ersten Überblick über mögliche Regressoren und deren Güte auf Indikatoren zu bekommen. Theoretisch ist es möglich, direkt eine Zeitreihe bei Google Correlate einzulesen und zu dieser die 100 am höchsten korrelierenden Wörter zu finden. Praktisch funktioniert dies aufgrund einer internen Fehlermeldung jedoch nicht. Somit müssen manuell Wörter ausgewählt werden, zu diesen über Goolge Trends eine Zeitreihe erstellt und mit einer linearen Regression wie im Punkt 4.2.2 beschrieben, diese nach ihrer Güte überprüft werden.

Als Vergleich wurde eine zweite Methode, das BSTS-Modell, beschrieben und auf den gleichen Datensatz angewendet. Auch hier wurde alles in *R* programmiert und ein von

Scott (2015) entwickeltes Package verwendet. Zu erwähnen ist, dass das *bsts* Package nicht mit täglichen oder auch wöchentlichen Werten rechnen kann. Lediglich monatliche Unterteilungen führten zu einem Output der Ergebnisse. Im direkten Vergleich der möglichen Prädiktoren mit dem linearen Regressionsmodell wird in beiden dem Wort „down economy“ ein positiver Zusammenhang auf den FRM unterstellt (Siehe Tabelle 4.1).

Interessant bei diesem Vergleich ist, dass im Datensatz zur Arbeitslosenquote die Ergebnisse nicht so eindeutig sind. Im BSTS-Modell hat die Variable „mini Job“ ein negatives Vorzeichen, im linearen Regressionsmodell jedoch ein positives. Intuitiv wäre ein positiver Zusammenhang zwischen der Anzahl der Suchanfragen zu „mini Job“ und der Arbeitslosenquote zu vermuten. Dies liegt daran, dass ein Minijob zusätzlich zur Beziehung von Arbeitslosengeld I ausgeübt werden darf. Dass die Kategorisierung im BSTS-Modell für diesen Prädiktor vielleicht nicht ganz zutreffend ist, zeigt auch die Abbildung 4.13. Hier wird durch Hinzunahme dieser Variable der linke Teil (siehe blauer Ausschlag) der Zeitreihe unterschätzt und der rechte Teil (gelber Ausschlag) überschätzt. Erst die Einbindung weiterer Prädiktoren gleichen dies wieder aus und die Zeitreihe kann durch dieses Modell sehr gut abgebildet werden.

Zumindest für die Vergangenheit lassen sich Zusammenhänge, wenn auch indirekt, zwischen Suchbegriffen und Konjunkturindikatoren feststellen. Die Arbeitslosenquote stimmt je Bundesland in über 60% der Fälle mit der Anzahl der Suche nach „Arbeitsamt“ überein. Auch die Korrelationen zu verwandten Begriffen wie „Ausbildungsstellen“ oder „Jobbörsen“ zeigt, dass solche Datenmengen nicht völlig unstrukturiert und zufällig sind.

Die Herausforderung wird sein, mehr und bessere Daten zu beschaffen. Diese richtig zu strukturieren und auch durch statistische Methoden zu interpretieren. Wenn auch nicht öffentlich zugänglich, gibt es heute mehr Daten als jemals zuvor.

> *„When you search on Google or Bing, your queries and subsequent clicks are recorded. When you shop on Amazon or eBay, not only every purchase, but every click is captured and logged. When you read a newspaper online, watch videos, or track your personal finances, your behavior is recorded. The recording of individual behavior does not stop with the internet: text messaging, cell phones and geo-locations, scanner data, employment records, and electronic health records are all part of the data footprint that we now leave behind us.“*
> (Einav and Levin, 2013)

Literatur

Altissimo, F., Cristadoro, R., Forni, M., Lippi, M., and Veronese, G. (2010). New eurocoin: Tracking economic growth in real time. *The review of economics and statistics*, 92(4):1024–1034.

Ammann, M., Frey, R., and Verhofen, M. (2014). Do newspaper articles predict aggregate stock returns? *Journal of Behavioral Finance*, 15(3):195–213.

Arribas-Bel, D. (2014). Accidental, open and everywhere: Emerging data sources for the understanding of cities. *Applied Geography*, 49:45–53.

Askitas, N. and Zimmermann, K. F. (2009). Google econometrics and unemployment forecasting. *German Council for Social and Economic Data (RatSWD) Research Notes*, (41).

Banerjee, A., Marcellino, M., and Masten, I. (2014). Forecasting with factor-augmented error correction models. *International Journal of Forecasting*, 30(3):589–612.

Bartenhagen, C. (2013). Rdrtoolbox: a package for nonlinear dimension reduction with isomap and lle.

Biau, O. and D'Elia, A. (2009). Euro area gdp forecasting using large survey datasets. Zuletzt besucht: 02.10.2015, Availabile via: http://unstats.un.org/unsd/nationalaccount/workshops/2010/moscow/AC223-S73Bk4.PDF.

Borke, L. and Härdle, W. K. (2015). Q3-D3-LSA. *SFB 649 Discussion paper (forthcoming)*. Humboldt Universität zu Berlin.

Castle, J. L., Qin, X., and Reed, W. R. (2009). How to pick the best regression equation: A review and comparison of model selection algorithms. Technical report.

Choi, H. and Varian, H. (2012). Predicting the present with google trends. *Economic Record*, 88(s1):2–9.

De Mol, C., Giannone, D., and Reichlin, L. (2008). Forecasting using a large number of predictors: Is bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146(2):318–328.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs*, 41(6):391–407.

DeJong, D. N. and Dave, C. (2011). *Structural macroeconometrics*. Princeton University Press.

Doornik, J. A. and Hendry, D. F. (2015). Statistical model selection with "big data". *Cogent Economics & Finance*, 3(1):1045216.

Dumbill, E. (2012). What is big data. http://orm-atlas2-prod.s3.amazonaws.com/pdf/e11376d1c19a651736042656f2aae705.pdf. Zuletzt besucht: 20.08.2015.

Economist (1998). The Recession Index. *The Economist - Print Edition*, Dec 12. http://www.economist.com/node/179331 Zueltzt besucht: 05.08.2015.

Einav, L. and Levin, J. D. (2013). The Data Revolution and Economic Analysis. Working Paper 19035, National Bureau of Economic Research.

Foulkes, A. S. (2009). *Applied statistical genetics with R: for population-based association studies*. Springer Science & Business Media.

Geiß, J. and Klein-Bering, J. (2003). Latent semantic indexing - tutorial. Universität Heidelberg, Lehrstuhl für Computerlinguistik. Hauptseminar: Information Retrieval Leitung: PD Dr. Karin Haenelt.

Giovannelli, A. (2012). Nonlinear forecasting using large datasets: Evidences on us and euro area economies.

Google (2011). Durchschnittliche dauer der onlinerecherche bis zur kaufentscheidung nach segment im jahr 2010 (in tagen). http://de.statista.com/statistik/daten/studie/182216/umfrage/dauer-der-onlinerecherche-bis-zur-kaufentscheidung-nach-segmenten/. In Statista - Das Statistik-Portal. Zuletzt besucht: 09.08.2015.

Google (2015). Trends help. https://support.google.com/trends/answer/4365533?hl=en. Zuletzt besucht: 20.08.2015.

Grace, G. H. and Desikan, K. (2015). Experimental estimation of number of clusters based on cluster quality. *arXiv preprint arXiv:1503.03168*.

Härdle, W. K., Duan, J.-C., and Gentle, J. E. (2011). *Handbook of computational finance*. Springer Science & Business Media.

Hastie, T. (2004). Boosting, random forest and bagging. Stanford University. Online: http://jessica2.msri.org/attachments/10778/10778-boost.pdf, Zuletzt besucht: 09.10.2015.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning.

Hendry, D. and Doornik, J. A. (2014). Statistical model selection with'big data'. Technical report.

Hothorn, T., Hornik, K., Strobl, C., and Zeileis, A. (2015). Party: A laboratory for recursive partytioning.

Howard, J. and Bowles, M. (2012). The Two Most Important Algorithms in Predictive Modeling Today. http://strataconf.com/strata2012/public/schedule/detail/22658. Zuletzt besucht: 15.08.2015.

IBM (2014). The four v's of big data. http://www.ibmbigdatahub.com/infographic/four-vs-big-data. Zuletzt besucht: 15.08.2015.

Iselin, D. and Siliverstovs, B. (2013). Using newspapers for tracking the business cycle: A comparative study for germany and switzerland. *KOF Swiss Economic Institute Working Paper*, (337).

Ishwaran, H., Rao, J., and Kogalur, U. (2013). spikeslab : Prediction and variable selection using spike and slab regression. R package version 1.1.5.

Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. (2010). Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11:517–553.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Lewis, R. J. (2000). An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California*, pages 1–14.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Lindh-Knuutila, T. (2014). *Computational Modeling and Simulation of Language and Meaning: Similarity-Based Approaches*. Aalto University publication series DOCTORAL DISSERTATIONS, 49/2014.

Madden, S. (2012). From Databases to Big Data. *IEEE Internet Computing*, (3):4–6.

Mao, Y., Balasubramanian, K., and Lebanon, G. (2010). Dimensionality Reduction for Text using Domain Knowledge. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 801–809. Association for Computational Linguistics.

McLeod, A. and Xu, C. (2015). bestglm: Best subset glm.

Mohebbi, M., Vanderkam, D., Kodysh, J., Schonberger, R., Choi, H., and Kumar, S. (2011). Google correlate whitepaper. http://www.google.com/trends/correlate/whitepaper.pdf. Zuletzt besucht: 12.08.2015.

okugami79 (2013). R package okugami79 and googletrend. Online on GigHub. Zuletzt besucht: 12.08.2015.

Ouysse, R. (2013). Forecasting using a large number of predictors: Bayesian model averaging versus principal components regression. *UNSW Australian School of Business Research Paper*, (2013-04).

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Perlich, C., Provost, F., and Simonoff, J. S. (2003). Tree induction vs. logistic regression: A learning-curve analysis. *The Journal of Machine Learning Research*, 4:211–255.

Preis, T., Moat, H. S., and Stanley, H. E. (2013). Quantifying trading behavior in financial markets using google trends. *Scientific reports*, 3.

Press, G. (2013). A Very Short History Of Big Data. http://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/ Zuletzt besucht: 21.09.2015.

Sagiroglu, S. and Sinanc, D. (2013). Big data: A review. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 42–47.

Schumacher, C. (2007). Forecasting german gdp using alternative factor models based on large datasets. *Journal of Forecasting*, 26(4):271–302.

Schumacher, C. and Breitung, J. (2008). Real-time forecasting of german gdp based on a large factor model with monthly and quarterly data. *International Journal of Forecasting*, 24(3):386–398.

Scott, S. L. (2015). bsts: Bayesian structural time series. *URL: http://CRAN.R-project.org/package=bsts*.

Scott, S. L. and Varian, H. R. (2014a). Bayesian variable selection for nowcasting economic time series. In *Economic Analysis of the Digital Economy*. University of Chicago Press.

Scott, S. L. and Varian, H. R. (2014b). Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5(1-2):4–23.

Statista-Das-Suchmaschinenportal (2015). Suchmaschinenverteilung. http://de.statista.com/statistik/daten/studie/167841/umfrage/marktanteile-ausgewaehlter-suchmaschinen-in-deutschland/. Zuletzt besucht: 11.08.2015.

Stephens-Devidowitz, S. and Varian, H. (2015). A Hands-on Guide to Google Data. *Working Paper*. http://people.ischool.berkeley.edu/ hal/Papers/2015/primer.pdf Zuletzt besucht: 07.08.2015.

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566.

Therneau, T. M., Atkinson, B., Ripley, B., et al. (2015). rpart: Recursive partitioning. *R package version*, 3:1–46.

Trendafilov, N. T. (2014). From simple structure to sparse components: a review. *Computational Statistics*, 29(3-4):431–454.

Varian, H. R. (2014). Big data: New tricks for econometrics. *The Journal of Economic Perspectives*, pages 3–27.

Würdinger, S. (2015). Kaufentscheidung - Überzeugungskraft kommt aus dem Internet. *TNS-Infratest*.

A Abbildungen

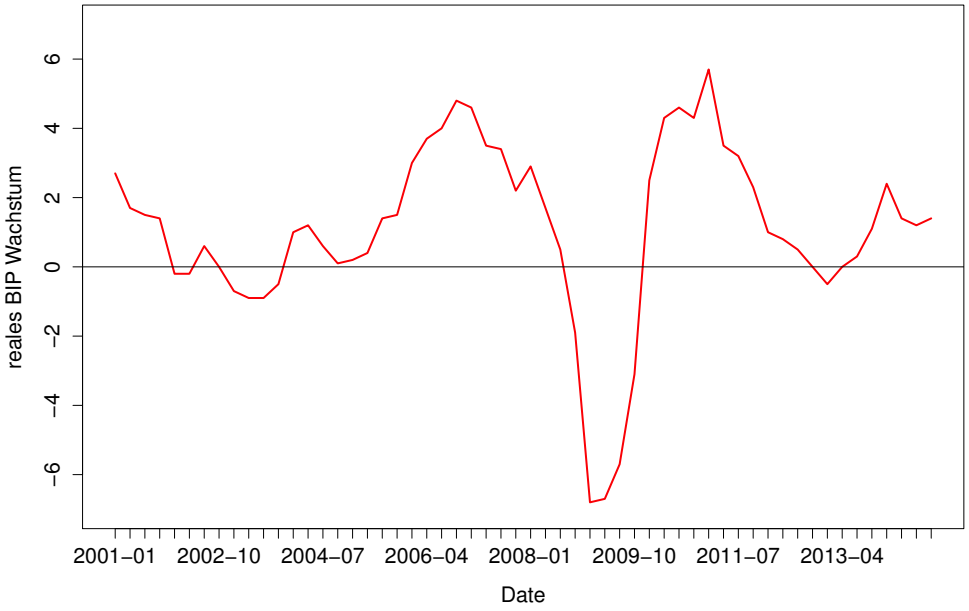

Abbildung A.1: Jahreswachstumsrate des BIP (quartalsweise). Quelle: Statistisches Bundesamt.

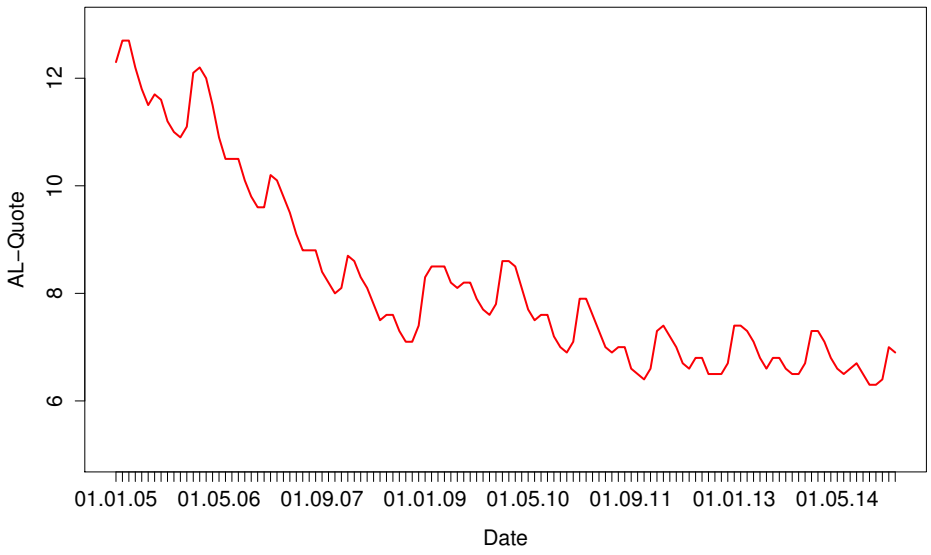

Abbildung A.2: Zeitlicher Verlauf der Arbeitslosenquote für Deutschland. Quelle: Statistisches Bundesamt.

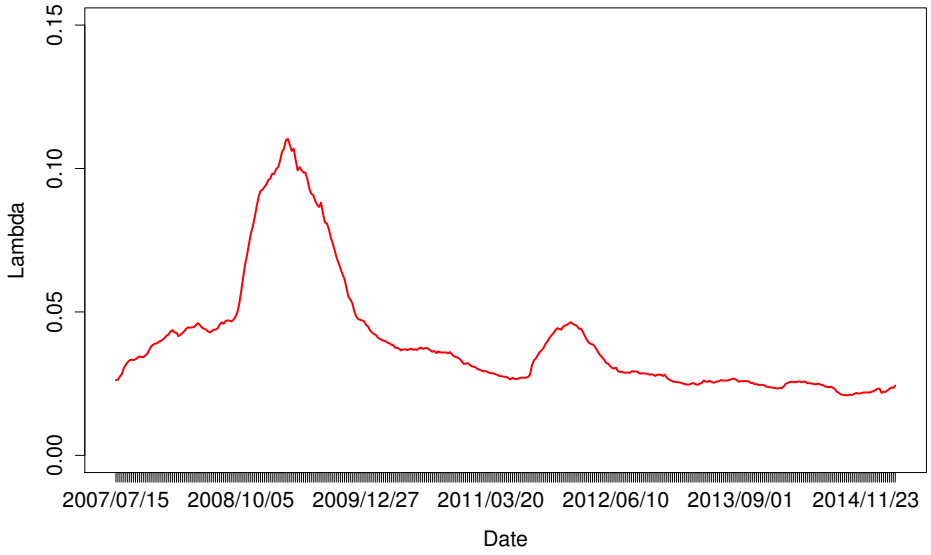

Abbildung A.3: Zeitlicher Verlauf des Financial Risk Meters. Quelle: SFB649, Lukas Borke.

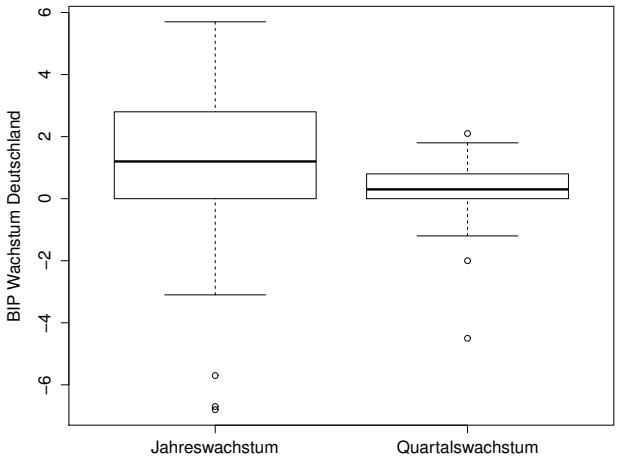

Abbildung A.4: Boxplot: BIP Deutschland

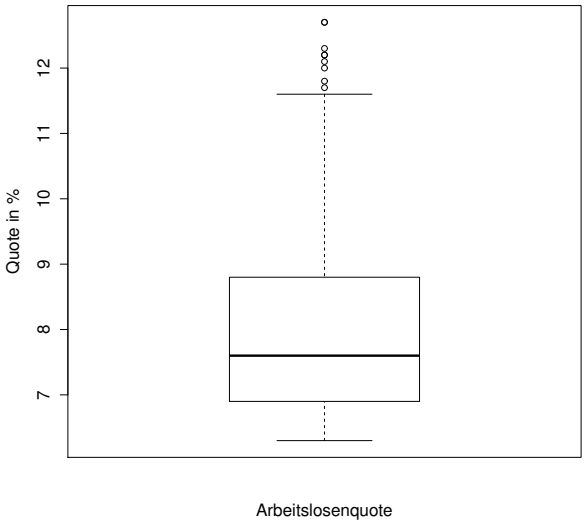

Abbildung A.5: Boxplot: Arbeitslosenquote

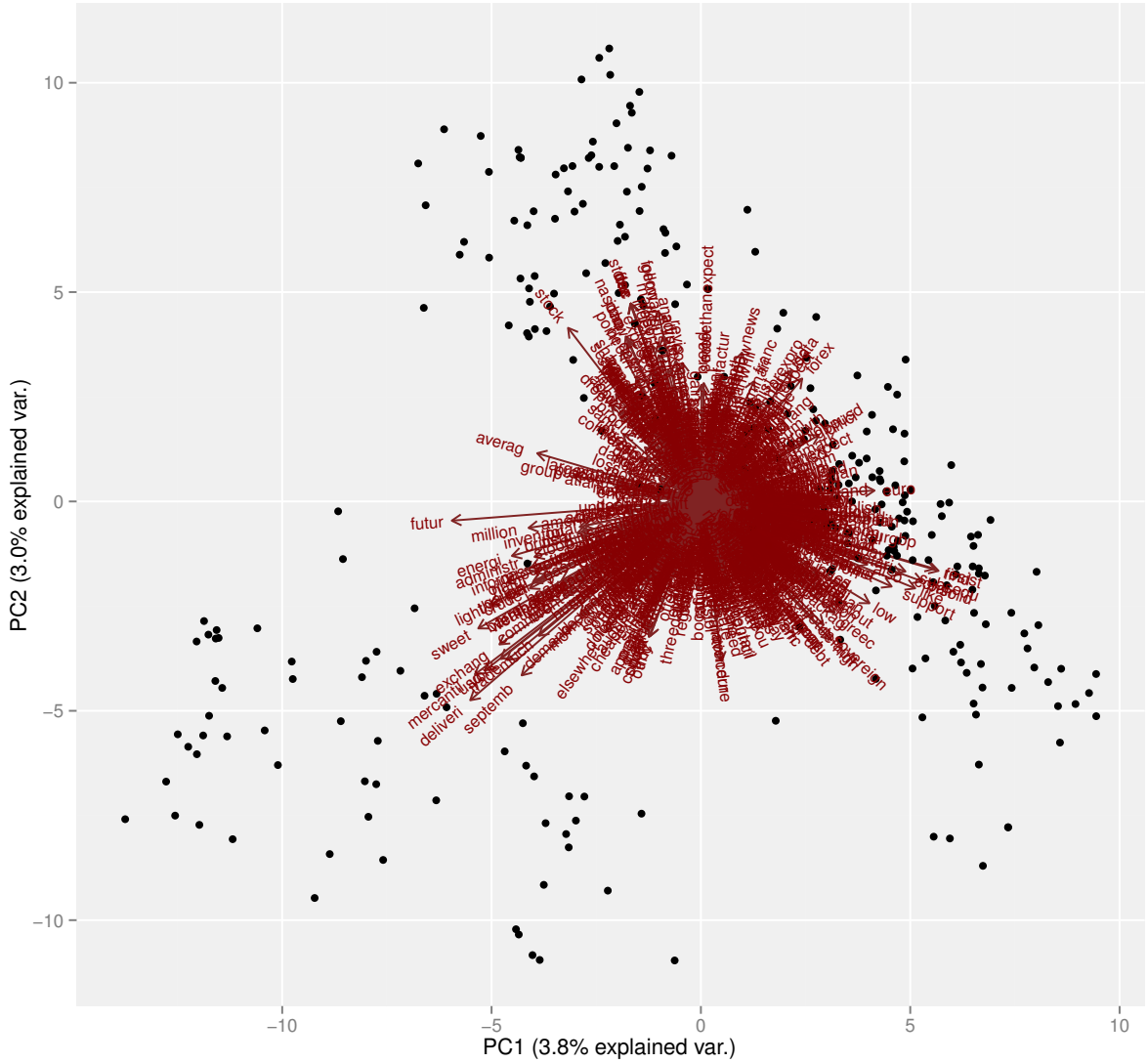

Abbildung A.6: Biplot von zwei Hauptkomponenten. Alle Terme der Stichprobe verwendet.
Quelle: Eigene Darstellung, Daten: Nasdaq Datensatz (Stichprobe von 300 Texten der „investing“ Artikel.)

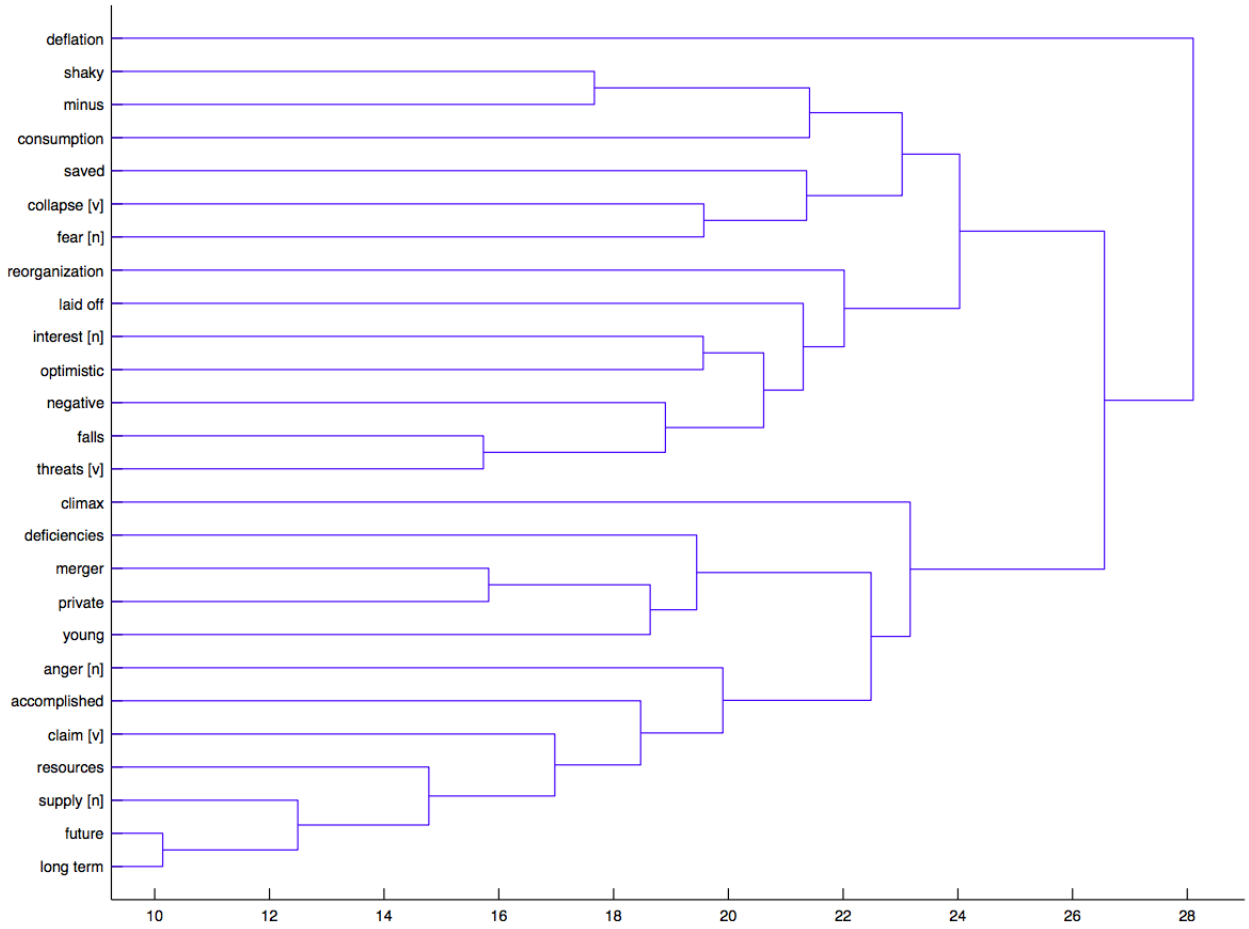

Abbildung A.7: Dendrogramm zur Clusterbildung aller signifikanten Wörter. Horizontale Achse beschreibt die Euklidische Distanz zwischen den Clustern. Quelle: Übernommen von Ammann et al. (2014).

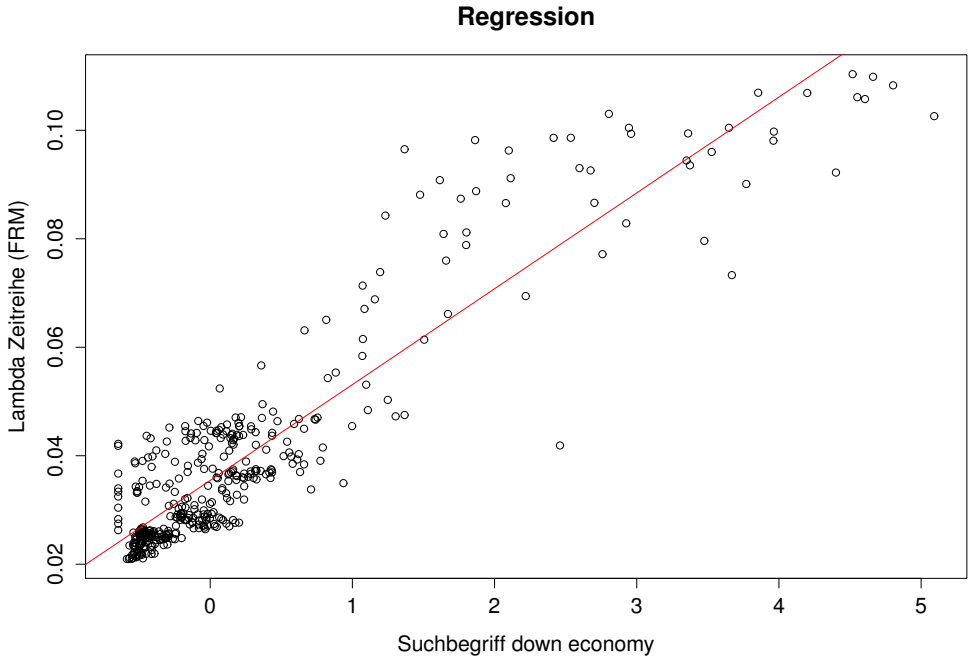

Abbildung A.8: Streudiagramm zwischen dem Suchbegriff „down economy“ und dem Financial Risk Meter (Lambda Zeitreihe).

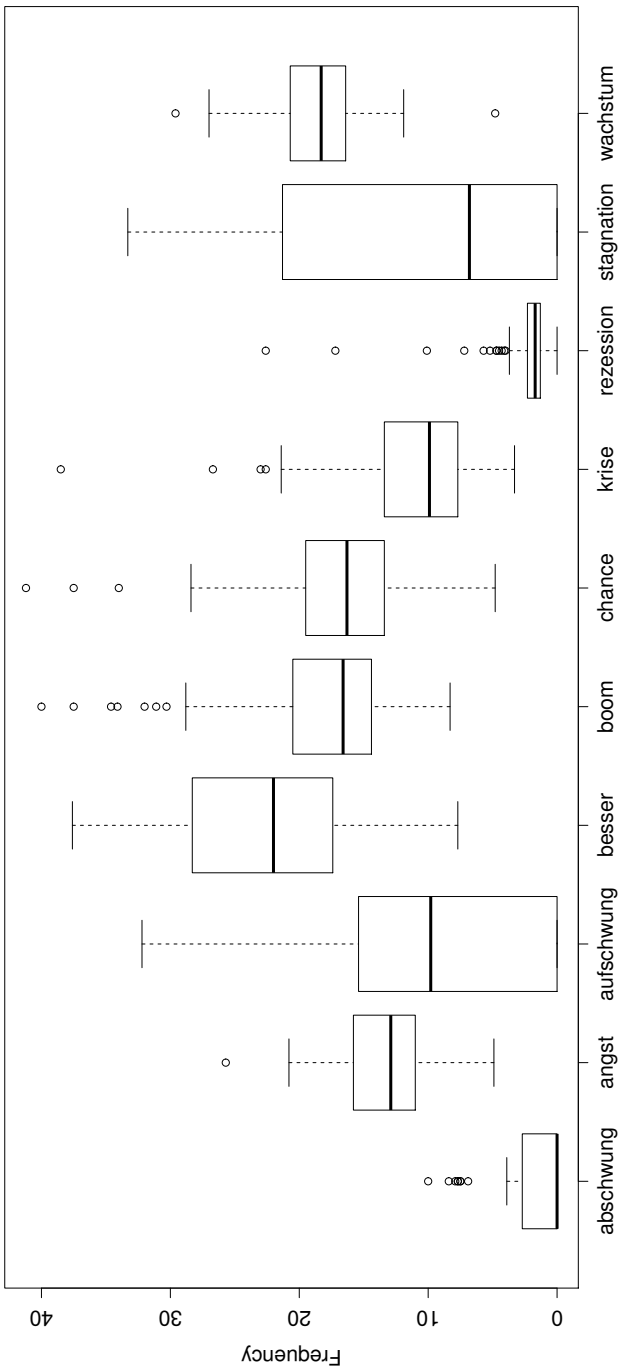

Abbildung A.9: Boxplots: Google Trends Wörter (DE)

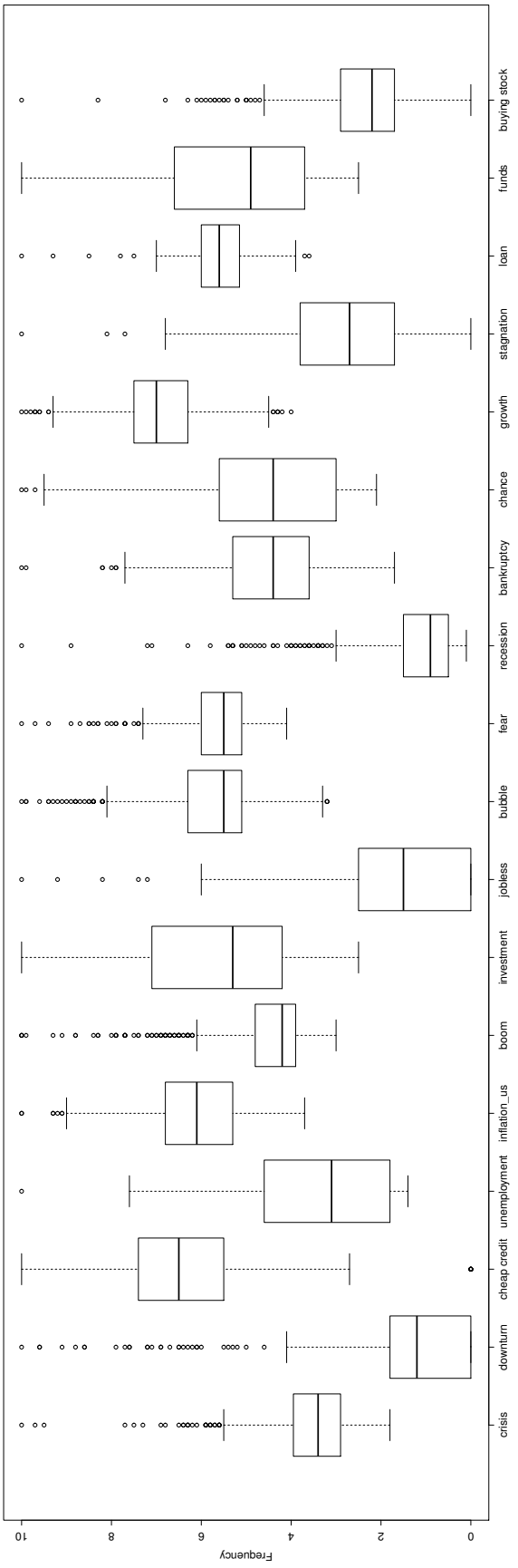

Abbildung A.10: Boxplots: Google Trends Wörter (U.S.)

B Tabellen

Tabelle B.1: Regression des Suchbegriffs „abschwung“ von Google auf das BIP von Deutschland.

	Dependent variable:
	BIP Deutschland
Suchbegriff: „abschwung“	-0.364^{***}
	(0.050)
Constant	0.716^{***}
	(0.115)
Observations	44
R^2	0.560
Adjusted R^2	0.549
Residual Std. Error	0.672 (df = 42)
F Statistic	53.417^{***} (df = 1; 42)
Note:	*p<0.1; **p<0.05; ***p<0.01

Tabelle B.2: Regression des Suchbegriffs „rezession“ von Google auf das BIP von Deutschland.

	Dependent variable:
	BIP Deutschland
Suchbegriff: „rezession“	-0.193^{***}
	(0.056)
Constant	0.723^{***}
	(0.180)
Observations	44
R^2	0.220
Adjusted R^2	0.201
Residual Std. Error	0.895 (df = 42)
F Statistic	11.822^{***} (df = 1; 42)
Note:	*p<0.1; **p<0.05; ***p<0.01

Hiermit erkläre ich, Daniel Jacob, dass ich die vorliegende Arbeit allein und nur unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.
Die Prüfungsordnung ist mir bekannt. Ich habe in meinem Studienfach bisher keine Bachelorarbeit eingereicht bzw. diese nicht endgültig nicht bestanden.

Daniel Jacob

Berlin 21. Oktober 2015